Co-funded by the
Erasmus+ Programme
of the European Union

ROMUAS

KILLS
RDUINO
SING
MACHINES
OUTDATED
ECOVER

C1

Argument

Expert program toolkit

PR1

1) Argument

The whole community faces many challenges in today's globalised society, both due to digitalization and climate change. All this contributes to the rapid change of the labour market, as well as the skills and abilities required. It is necessary to strengthen human resource skills, competitiveness and employability, prioritising education as the engine of society's development.
Nowadays, intelligence, innovation and creativity have become relevant reference points. They represent a guarantee of future prosperity for the whole world. We live in a world where they demand not only high value products and markets, but, increasingly, high-value skills. To ensure a knowledge society, excellence should be ensured at all stages of the educational process, to permanently update the skills base of the population according to needs, and to create a social environment, economic and regulatory environment that can stimulate research, creativity and innovation.
The purpose of the Expert program toolkit is to create a training material that provides the target groups of the project with high quality knowledge and skills needed to modernise existing machineries in order to make them “smart” and more adapted to the current requirements of the manufacturing industry.
The toolkit consists of:

1) a theoretical part and a practical part containing tutorials and videos on how to recover outdated machines (devices) using Arduino technology
2. a presentation of social innovation describing briefly the development of technology and then focusing on social skills and the impact of technology on those skills ending with interactive methods of developing them
3. a theoretical presentation of entrepreneurial competences and interactive methods to develop them
4. considerations on green skills, circular economy, advantages and benefits of refurbishing outdated devices and using them and interactive methods of developing the green skills

The people that will follow the first part of the program will have the chance of learning through relevant examples how to recover outdated machines that would otherwise be discarded using Arduino technology. The suggested examples would make them aware of a few possibilities of the uses of Arduino components and software and would spark their interest and creativity since the use of the aforementioned technology is limitless.
Those that will follow the second part will have the chance of finding some considerations on the development of technology and by answering the questions in the questionnaires and interpreting the results will have the chance of evaluating the level of their social skills and spotting the aspects that need to be improved.

Co-funded by the
Erasmus+ Programme
of the European Union

Co-funded by the
Erasmus+ Programme
of the European Union

ROMUAS
KILLS
RDUINO
SING
MACHINES
OUTDATED
ECOVER

Recover updated machines (devices) using Arduino technology

C2

Expert program toolkit

PR1

2) Recover updated machines (devices) using Arduino technology

I. Theoretical part

Arduino Basics and Programming Software

What is Arduino?

In general, Arduino is an electronic prototyping platform, which integrates the popular AVR microcontroller, and allows the development of interactive control systems. It is a low-cost device and accessible to everyone. Some features are:

- It does not require an external programmer;
- Interacts seamlessly with a dedicated compiler;
- Is compatible with numerous expansion cards (e.g. motor controllers, displays, modules, sensors, etc.).

However, the real power of the Arduino lies in its C/C++ based programming language.

Programming is done through bookstores, thanks to which even the creation of a complicated program is within the reach of a beginner programmer.

The Benefits of Arduino

The project started in 2005, in Italy. Since then it has gathered a multitude of followers and users. From the beginning, Arduino was designed for people who had little knowledge of microcontroller programming. Its excellent software, friendly syntax and low price have made Arduino extremely popular.

The community built around this project is huge. This brings many benefits. From a beginner's point of view, the most important are the following:

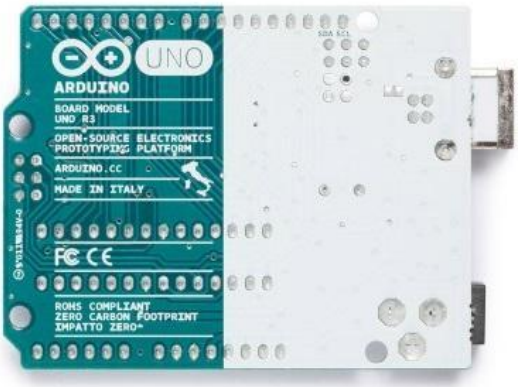
Co-funded by the Erasmus+ Programme of the European Union

- A large number of ready-made solutions. There is a plethora of Arduino projects already created. If you think you have created something "new" and interesting, it is likely that one of the users has already done so and published it on the Internet;
- The popularity of the platform means that there are countless varieties of boards and extensions created by different manufacturers - we will talk about this in a moment;
- A large number of users makes it easy to find help when you get stuck somewhere in the project.

Arduino - Choosing Hardware

Arduino is an Open Hardware platform. This means that all the materials needed to create your own device are available. For this reason, there are lots of Arduino-compatible boards.

Currently there are more than 20 original models available. In any good electronics store you will find several of these models. For the purpose of the course we have chosen the most popular board - Arduino UNO R3.



Arduino UNO R3 Constitution

The heart of the Arduino is the popular Atmel 8-bit microcontroller, the AVR ATmega328 that operates at 16MHz. Working at such a frequency, very simply, means that the microcontroller is capable of performing 16 million operations per second!

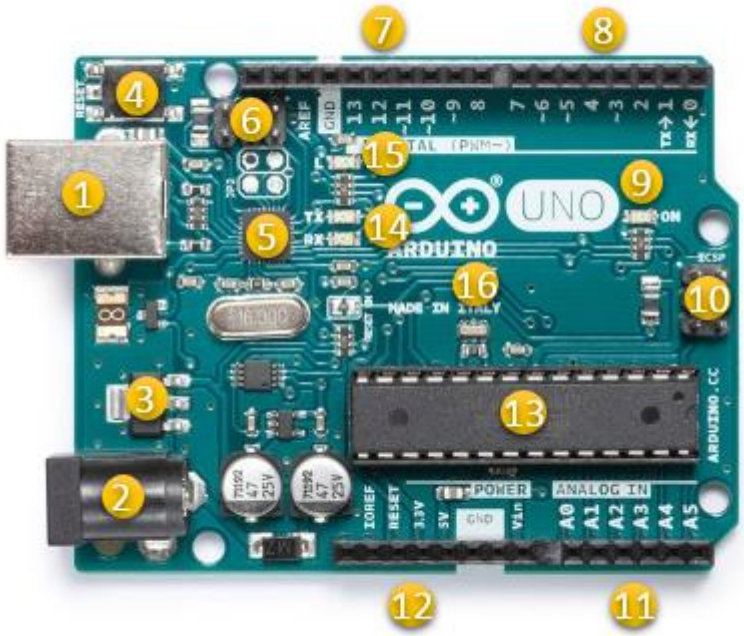
The connectors located on the sides of the board are the outputs of the most important signals. There we find 14 programmable digital inputs/outputs (I/O). Six of them can be used as PWM outputs (e.g. motor control) and the other 6 as analog inputs. We also find a reset button and a connector for the board's power supply.

The Arduino can be powered in several ways. The most popular methods are:

- Power supply via USB cable;
- Power supply via external power source (batteries or plug-in).

Co-funded by the Erasmus+ Programme of the European Union

The most important constituents are marked in the following image:



1. USB connector - used for power, programming, and communication with the computer;
2. DC Jack connector - used for power supply (7V to 12V recommended);
3. Voltage Stabilizer - transforms the input voltage to 5V;
4. Reset Button - resets the Arduino board;
5. Microcontroller - responsible for communication with the computer via USB;
6. Programming terminals for microcontroller #5
7. Digital I/O pins (8 to 13) and Digital GND (ground);
8. Digital I/O pins (3 to 7) and TX and RX (serial output and serial input respectively);
9. LED - indicates if the board is powered on;
10. Serial Programming Pins - allows you to program the microcontroller (pin #13) via serial communication;
11. Analog Input Pins (0 to 5);
12. Board Power and Reset Pins;
13. AVR ATmega328 microcontroller - the heart of the Arduino board;
14. LEDs - indicate the transmission from/to the computer;
15. LED - connected to pin #13, is available to the user;
16. 16MHz crystal - causes the microcontroller to operate at a frequency of 16MHz.

Co-funded by the Erasmus+ Programme of the European Union

Clones vs. Arduino Original

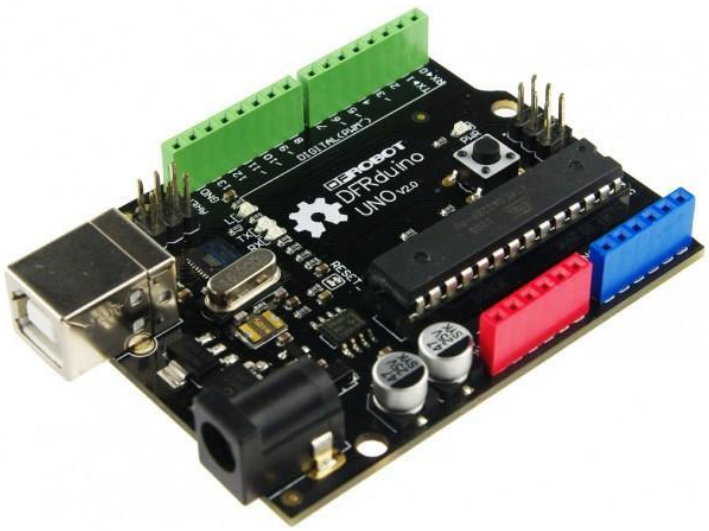
As mentioned, Arduino is an open hardware platform. This means that everyone can create their own Arduino or design a similar board. Boards identical to Arduino are colloquially called clones.

Clones can be divided into two types:

- Integral fakes that imitate the original;
- Arduino compatible boards.

You can take the risk and purchase a counterfeit for half the price (or less). But, the truth is, they are made with lower quality materials, this can cause malfunctions and problems more easily. As for compatible boards, they can be just as efficient as the original ones.

There are also other cheaper boards that are compatible with Arduino, but the manufacturers do not pretend to be original. These versions are produced, for example, by DFrobot, which sells its boards under the name DFRduino.



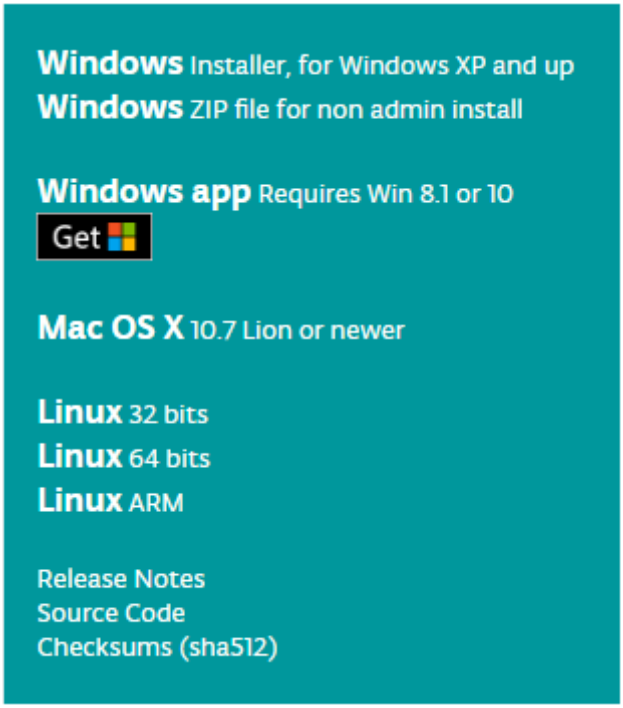
DFRduino is produced by a reliable manufacturer, so you can easily purchase it. Arduino compatible products can be quickly identified by using the suffix -uino.

Co-funded by the
Erasmus+ Programme
of the European Union

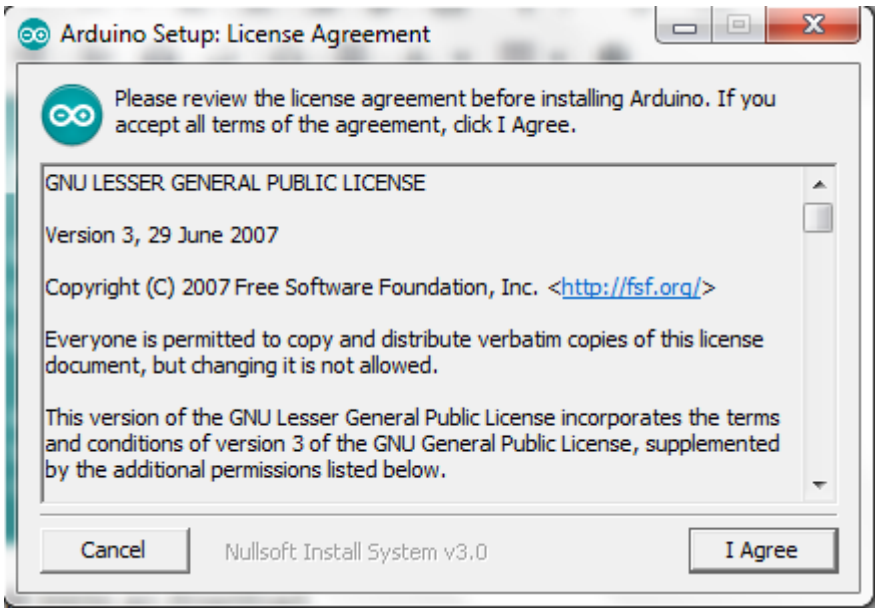
Installing the Arduino IDE

Before you start programming, appropriate software should be installed. You can download the latest Arduino IDE from the official Arduino website. The software takes about 90MB.

Installation is basic. To get started you will need to choose your operating system.

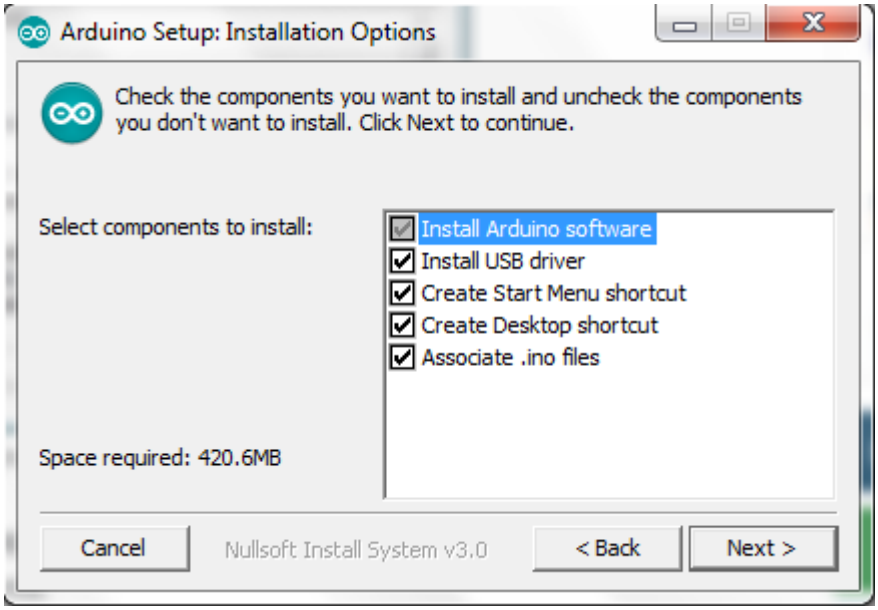


After the download is done, we will start the installation. First you will have to accept the product license:

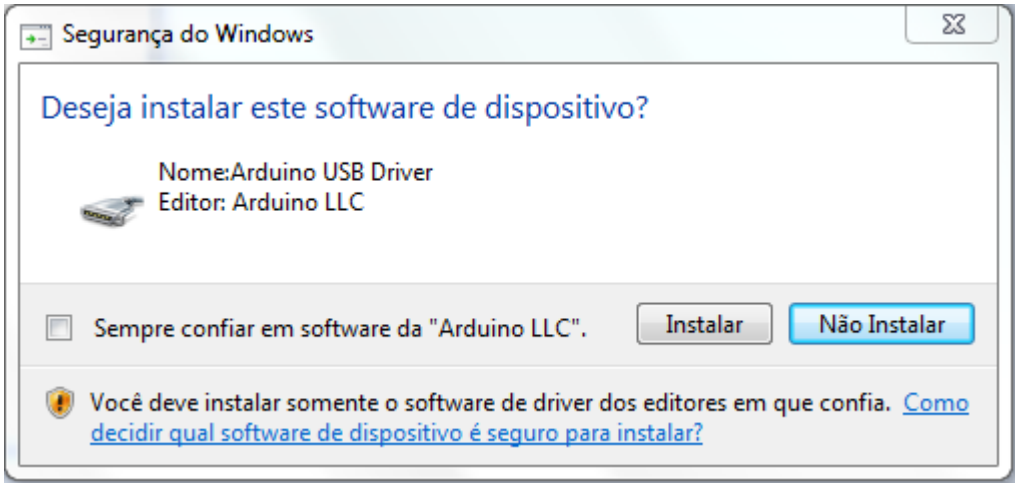


Co-funded by the Erasmus+ Programme of the European Union

Then click Next. Meanwhile pay attention to the components to install:

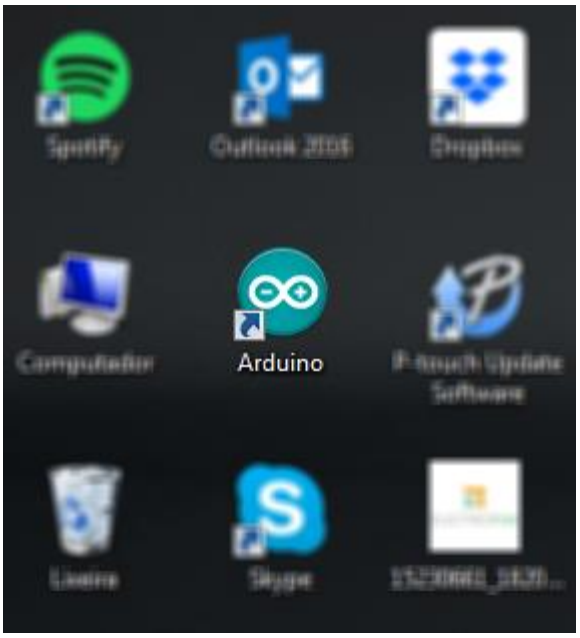


If you choose to install the USB driver (recommended), you will receive a warning at the end of the installation that looks like this:



Click on "Install". You don't need to worry about anything. Then click Close and the software is installed. An icon like the following should appear on your desktop:

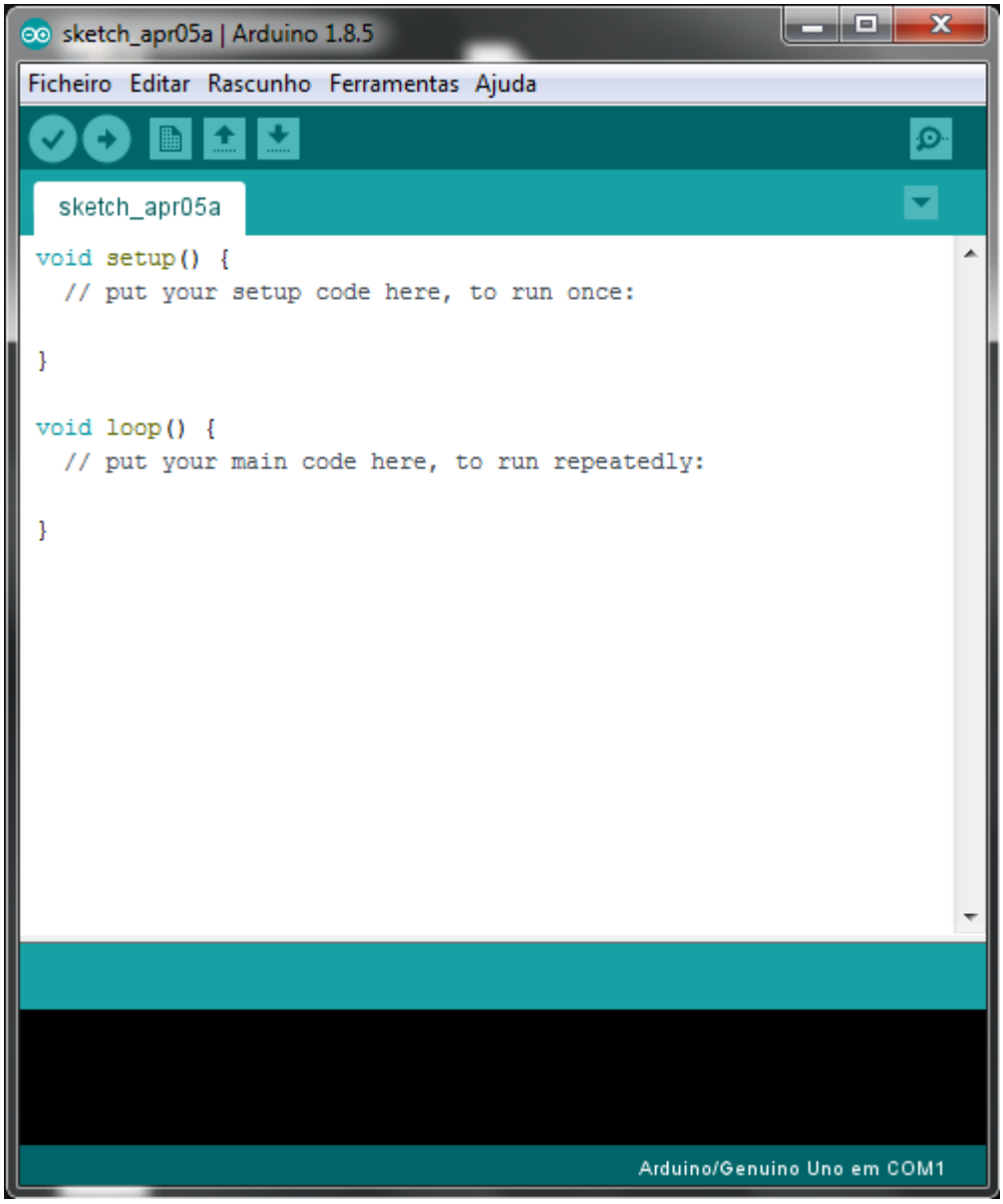
Co-funded by the Erasmus+ Programme of the European Union



If the software has been properly installed, when you open the IDE, a welcome image like this should appear:

Co-funded by the
Erasmus+ Programme
of the European Union

After a few seconds, the editor:



The Basics of Arduino Programming

In C language, all the instructions we want to give, we put in the main function (more on functions below):

```
1  int main() {
2
3  //Conteúdo do Programa
4
5  }
```

The symbol "//" indicates a comment. It is a one-line piece of information that helps people understand the program. During compilation, all comments are omitted. If you want to write a longer comment, you must enclose it in "*/".

Co-funded by the Erasmus+ Programme of the European Union

In Arduino, there are aspects that are simplified. There are two functions: one of them executes the instruction once, the other executes the defined instruction in a loop. Let's see:

```
void setup() {
//Instruções que são executadas apenas uma vez
}

void loop() {
//Instruções que são executadas em loop
}
```

In practice, the first function usually contains the settings. For example: setting the pins as inputs or outputs. In this function, after you switch on the board, you will perform actions that are only supposed to happen once.

In the second function you put the code that you want to execute all the time (in loop).

Functions - what they mean?

The codes can be written by you, or you can use a pre-made one, provided by programmers or geeks who share their own code.

There is a concept of function in the C language. A function, in programming languages, is a block (list) of certain commands taken from the main code, whose performance provides a result.

Each function can take several arguments and send a result. The programmer can determine which values will be the result and the input data. Each function has its own type of result sent (prefixes e.g. int, string, etc.) - it can be a number, a sign or something else. There is also a specific type of function - it does not send any value (prefix void).

Let's focus on the main functions of Arduino programs.

```
void setup() {
}
```

The setup procedure, as we have already seen, is intended for executing the set up instructions only once. As the name suggests, it is intended primarily for general setup. Here, the processor is started, peripherals are configured, etc.

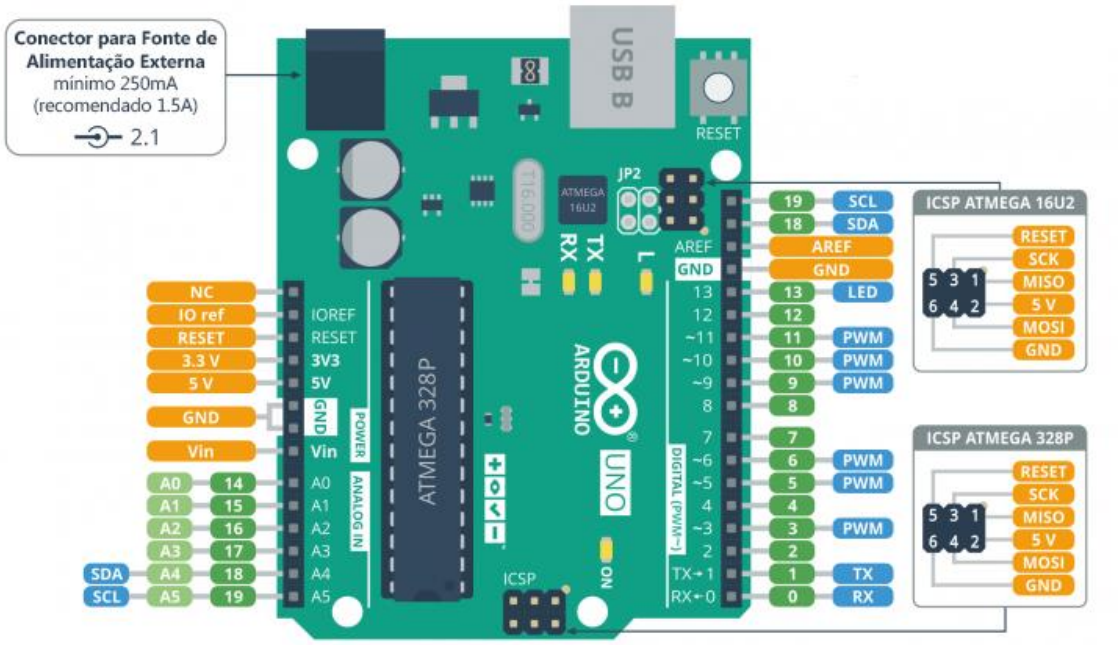
```
void loop() {
}
```

The loop function (procedure) is, as the name implies, an infinite loop. It is suitable for statements that must be executed all the time.

Co-funded by the Erasmus+ Programme of the European Union

Arduino UNO - pinout

Through certain connectors, you can connect external elements to the Arduino, such as: LEDs and buttons. However, before we get to that part, we need to know the description of the existing pins/connectors on the board. Below you can see a scheme with the main pins of the Arduino UNO:



In dark green (#0 to #19) are indicated the digital input/output (I/O) pins. When used as outputs, we can define them as either 0V (logic level 0, LOW) or 5V (logic level 1, HIGH). When they are configured as inputs, they are able to detect whether the pin in question has a voltage of 0V or 5V.

The analog inputs (A0-A5) are highlighted light green. These are unique pins that allow you to measure voltage (0-5V). As you can see, the numbering of these pins coincides with the universal pins (#14 to #19). Working in analog mode is an additional function of them.

At blue, pins with alternative functions have been highlighted. This means that, in addition to being normal I/O pins, they can perform more complex functions. Basic explanation:

- **SDA,** SCL - I2C buses used, for example, for communication with more advanced sensors. There are two SDA pins and two SCL pins, in the lower left and upper right corner of the board;
- **TX, RX** - UART interface, used mainly for communication with the computer;
- **PWM** - outputs on which you can generate a rectangular (variable) signal. It is a very useful function, for example, in servo control;
- **LED** - LED permanently installed in the Arduino, which is directly connected to pin #13.

In orange are outputs that are not programmable. These are mainly responsible for powering the system.

The connectors indicated in the image as ICSP are used for direct programming of two microcontrollers, which are located on the Arduino UNO board. These connectors are used in very specific cases.

Co-funded by the Erasmus+ Programme of the European Union

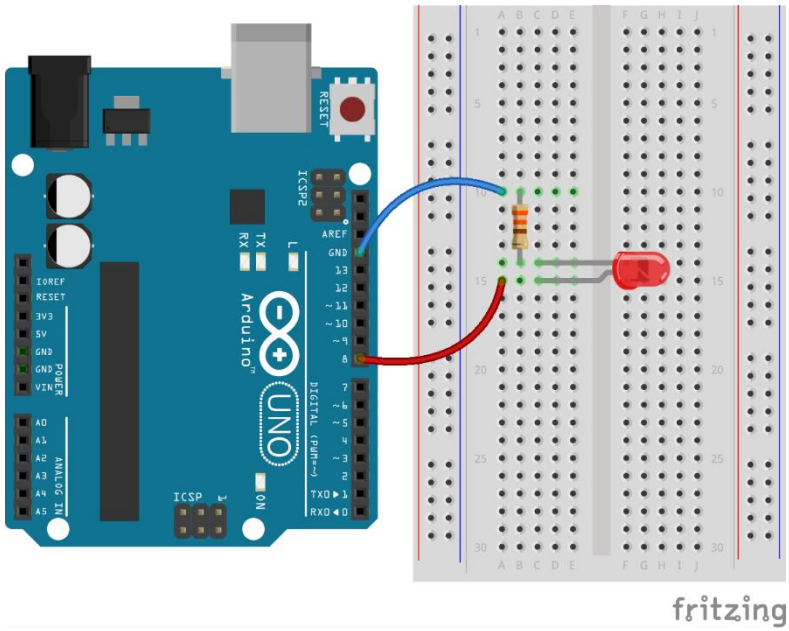
Practical Exercise Outputs - LED

Let's start the practice with something very simple: connecting an LED. According to the previous description, any I/O pin can be used for this connection. For now, choose digital output #8. Digital output is the output that can be set to one of two states (HIGH or LOW). In the case of the Arduino, this will be 5V or 0V.

Material required:

- - 1x Arduino UNO board and USB cable;
- - 1x Breadboard;
- - 1x LED;
- - 1x 330Ω resistor;
- - 2x Male-male jumper cables.

The system must be connected according to the schematic shown. The LED is connected in series with a resistor (330RΩ). Next, connect the longest pin of the LED (anode) to the same column as the jumper connected to pin #8. The second pin, should be in the same column as one leg of the resistor. The other leg should be in the same column as the jumper connected to the ground pin (GND). There are 3 GND pins on the board, you can choose any one.



Programming the LED inclusion is very simple. Connect the Arduino to the computer with the USB cable. Open the Arduino IDE and write the code below. Then upload it to the board.

```
1  void setup() {
2  pinMode(8, OUTPUT);
3  digitalWrite(8, HIGH);
4  }
5
6  void loop() {
7  }
```

Co-funded by the Erasmus+ Programme of the European Union

The function **pinMode** (Pin, Mode) allows you to select whether the pin is an input or an output. The pin is an integer between 0 and 13 and the mode can be:

- INPUT
- OUTPUT
- INPUT_PULLUP.

If we want to control an output, we use the OUTPUT mode.

Thanks to this setting you can set the logic state at the output and thus activate the LED. The function **digitalWrite** (Pin, Status) is used for this purpose. The status is a logical state that can be either HIGH or LOW.

In this example, the LED has already been connected to the ground pin, so the Arduino must reach a high state: digitalWrite (8, HIGH).

After setting the pin to a single state, its value will not change until you set a different value. Therefore, the above program will cause the LED to stay on all the time.

Practical Exercise Delays - LED Blinking

In this exercise we want to set the LED to blink. To do this, we need a new function to insert the delay. The wiring scheme is exactly the same as above. The code should look like this:

```
void setup() {
pinMode(8, OUTPUT); //Definir o pino 8 como saída
}

void loop() {
digitalWrite(8, HIGH); //Ligar o LED
delay(1000); //Esperar 1 segundo
digitalWrite(8, LOW); //Desligar o LED
delay(1000); //Esperar 1 segundo
}
```

In the loop function, the state is constantly alternated. Delays have been added to the program via the delay function. This function assumes a certain number of milliseconds to delay.

If you did not introduce the delays, the system would change state so quickly that it would be impossible to see the toggle with the naked eye. You can perform this experiment by setting 0ms as the delay.

Practical Input Exercise - Conditional (if) statement

It is often desired that the system reacts to external signals. This time we will connect the push button to the Arduino, in addition to the LED.

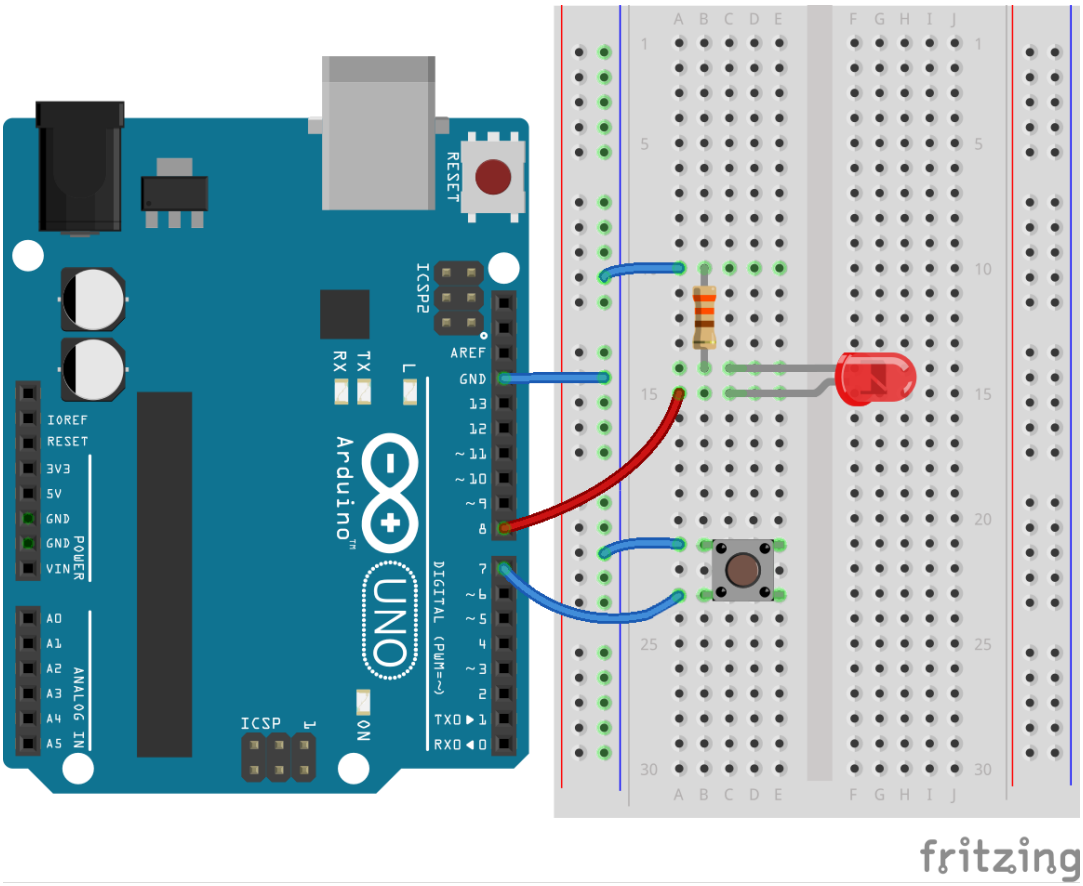
Material required:

- - 1x Arduino UNO and USB cable;

Co-funded by the Erasmus+ Programme of the European Union

- - 1x Breadboard;
- - 1x LED;
- - 1x Pushbutton;
- - 1x 330Ω resistor;
- - 5x Jumper cables.

The connection should be made according to the diagram below. On one side, the button has been connected to ground (minus) and on the other side to pin 7.



The goal is to create a program that turns on the LED when the button is pressed. The task is very simple, but we will insert something new - **conditional statements**.

We want the program to be permanently in one of two states - the LED is either on or off. At first, we need to read the logic state of the input pin of the button.

Remember the **INPUT_PULLUP mode** mentioned earlier. The first part of the name (input) obviously means input, while the second (pullup) suggests the inclusion of an internal resistor that checks the state of the switch. We will use this whenever we connect a button to the Arduino.

This requires the **digitalRead** (pin) function, which sends HIGH or LOW information, depending on the state. However, just reading the state of the input is not enough, we must make the program work depending on this information. Hence the conditional (if) statement. Thanks to this, you can execute a certain part of the code if certain requirements are fulfilled or if certain conditions are met. For example, if you press a button.

Co-funded by the Erasmus+ Programme of the European Union

```
[...]
void loop() {

if (condição) {
/* A instrução é executada em loop somente
quando a condição é cumprida*/
}

}
```

This function can easily be extended with a piece of code that will be executed only if the condition is not checked. The else statement is used for this.

```
[...]
void loop()
{

if ( condição) {
/* A instrução é executada em loop somente
quando a condição é cumprida */
} else {
/* A instrução é executada somente
quando a condição não é cumprida*/
}

}
```

By combining the knowledge you have gained, you can create a program that performs the proposed task. Review the code and upload it to the Arduino.

```
void setup() {
pinMode(8, OUTPUT); //LED como saída
pinMode(7, INPUT_PULLUP); //Botão como entrada
digitalWrite(8, LOW); //Desliga o LED
}

void loop()
{
if (digitalRead(7) == LOW) { //Se o botão for pressionado
digitalWrite(8, HIGH); //Ligar o LED
} else { //Se o botão não for pressionado
digitalWrite(8, LOW); //Desligar LED
}
}
```

Co-funded by the
Erasmus+ Programme
of the European Union

Example - Light Switch with "Timer"

In this example we want, given the previous example, the LED to be on for 10 seconds after pressing the button.

Can you write a suitable program? We hope so! If you have difficulties, you can always take a look at our code:

```
void setup() {
pinMode(8, OUTPUT); //LED como saída
pinMode(7, INPUT_PULLUP); //Botão como entrada
digitalWrite(8, LOW); //Desliga o LED
}

void loop()
{
if (digitalRead(7) == LOW) { //Se o botão for pressionado
digitalWrite(8, HIGH); //Ligar o LED
delay(10000); //Esperar 10 segundos
digitalWrite(8, LOW); //Desligar LED
}
}
```

Example - Traffic Light

The next example is a sequential traffic light system. The main goal is to write a program that, after the button is pressed, shows a correct sequence of lights. Let's assume the following cycle:

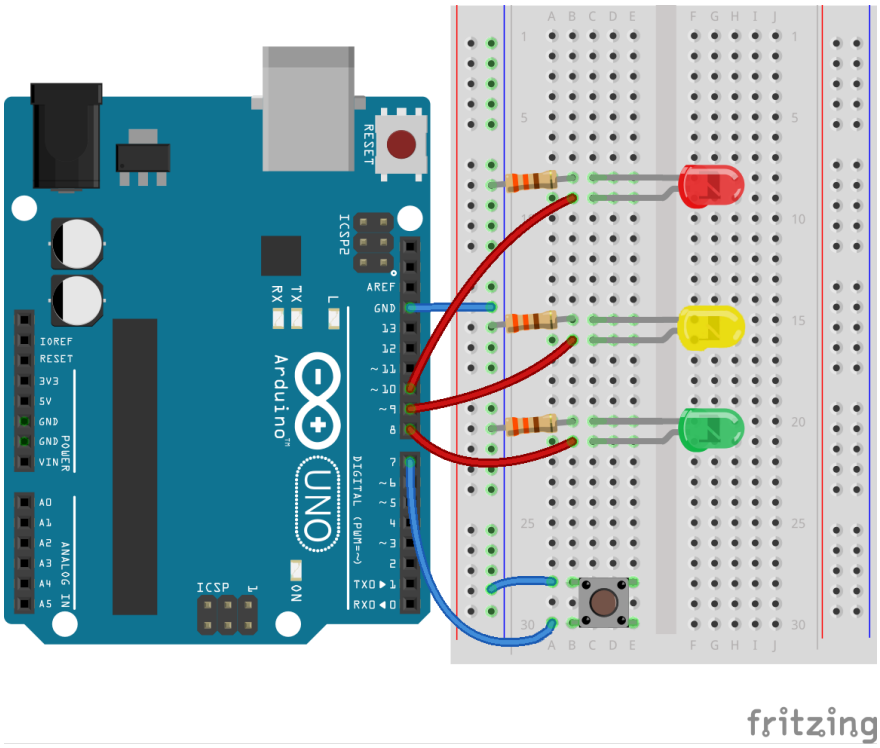
[…] Verde -> Amarelo -> Vermelho -> Amarelo -> Vermelho […]

Material required:

- 1x Arduino UNO and USB cable;
- 1x Breadboard;
- 1x Pushbutton;
- 1x Red LED;
- 1x Yellow LED;
- 1x Green LED;
- 3x 330Ω resistors;
- 6x Jumper cables.

When you press the button, the system should start the sequence. Let's do it in steps. First, turn on the 3 LEDs as shown below:

Co-funded by the Erasmus+ Programme of the European Union



We will prepare a program that serves only to configure the inputs and outputs.

```
void setup() {
pinMode(10, OUTPUT); //LED vermelho
pinMode(9, OUTPUT); //LED amarelo
pinMode(8, OUTPUT); //LED verde

pinMode(7, INPUT_PULLUP); //Botão

digitalWrite(10, LOW); //Desligar os LEDs
digitalWrite(9, LOW);
digitalWrite(8, LOW);
}
```

Now that the LEDs and the button are set up, let's write a program that changes the lights automatically, every 1 second. The complete sketch should look like this:

Co-funded by the
Erasmus+ Programme
of the European Union

```
void setup() {
pinMode(10, OUTPUT); //LED vermelho
pinMode(9, OUTPUT); //LED amarelo
pinMode(8, OUTPUT); //LED verde

pinMode(7, INPUT_PULLUP); //Botão

digitalWrite(10, LOW); //Desligar os LEDs
digitalWrite(9, LOW);
digitalWrite(8, LOW);
}

void loop()
{
digitalWrite(10, LOW); //Vermelho
digitalWrite(9, LOW); //Amarelo
digitalWrite(8, HIGH); //Verde

delay(1000); //Esperar 1 segundo

digitalWrite(10, LOW); //Vermelho
digitalWrite(9, HIGH); //Amarelo
digitalWrite(8, LOW); //Verde

delay(1000); //Esperar 1 segundo

digitalWrite(10, HIGH); //Vermelho
digitalWrite(9, LOW); //Amarelho
digitalWrite(8, LOW); //Verde

delay(1000); //Esperar 1 segundo

digitalWrite(10, HIGH); //Vermelho
digitalWrite(9, HIGH); //Amarelo
digitalWrite(8, LOW); //Verde

delay(1000); //Esperar 1 segundo
}
```

Upload the program to the Arduino and check that everything is working perfectly.

Co-funded by the
Erasmus+ Programme
of the European Union

Practical Exercise - While Function

So far, we have only used the main and mandatory loop (void loop function). Now it is time to learn about a loop that we can use inside our programs.

Now we will cover the **while** () loop, which works as long as a certain condition is met (true). Its operation is shown in the following code:

```
[...]
void loop()
{

while (condição) {

/* O código é executado em loop enquanto
a condição for verdadeira */

}

}
```

For clarity, the while function only executes the code that is between its {} curly braces. The rest of the code is not executed at that point.

Let's take advantage of the traffic light system that was set up earlier, to write a program that will flash one LED at a time when the button is pressed. This exercise is a more difficult task than the previous one.

Check out the code we propose:

```
void setup() {
pinMode(10, OUTPUT); //LED vermelho
pinMode(9, OUTPUT); //LED amarelo
pinMode(8, OUTPUT); //LED verde

pinMode(7, INPUT_PULLUP); //Botão

digitalWrite(10, LOW); //Desligar LED
digitalWrite(9, LOW);
digitalWrite(8, LOW);
}

void loop() {

while (digitalRead(7) == LOW) { //Quando o botão é pressionado
digitalWrite(10, LOW); //Vermelho desligado
delay(1000);
digitalWrite(10, HIGH); //Vermelho ligado
delay(1000);
}

}
```

Co-funded by the Erasmus+ Programme of the European Union

This time, the sequences should be displayed until the button is pressed. We assume that the button is pressed and released very quickly. The finished program should look like this:

```
void setup() {
pinMode(10, OUTPUT); //LED vermelho
pinMode(9, OUTPUT); //LED amarelo
pinMode(8, OUTPUT); //LED verde

pinMode(7, INPUT_PULLUP); //Botão

digitalWrite(10, LOW); //Desligar LED
digitalWrite(9, LOW);
digitalWrite(8, LOW);
}

void loop()
{
digitalWrite(10, LOW); //Vermelho
digitalWrite(9, LOW); //Amarelo
digitalWrite(8, HIGH); //Verde

while (digitalRead(7) == HIGH) {} //Quando o botão é pressionado

digitalWrite(10, LOW); //Vermelho
digitalWrite(9, HIGH); //Amarelo
digitalWrite(8, LOW); //Verde

while (digitalRead(7) == HIGH) {} //Quando o botão é pressionado

digitalWrite(10, HIGH); //Vermelho
digitalWrite(9, LOW); //Amarelo
digitalWrite(8, LOW); //Verde

while (digitalRead(7) == HIGH) {} //Quando o botão é pressionado

digitalWrite(10, HIGH); //Vermelho
digitalWrite(9, HIGH); //Amarelo
digitalWrite(8, LOW); //Verde

while (digitalRead(7) == HIGH) {} //Quando o botão é pressionado
}
```

In this case, the loop was used in a rather strange way. As you can see, there is nothing inside the curly braces! So how is the program working? This is because the program uses loops to stop.

How is it working?

1) We start illuminating the LEDs according to a sequence;
2) We enter the while () loop function, which is immediately below;
3) The curly braces are empty, so the program is always in a loop, doing nothing;
4) Only after the button is pressed (the condition becomes false) does the program exit the loop;
5) The next sequence is triggered and the situation repeats.

Let's now check the program in practice!!

Co-funded by the Erasmus+ Programme of the European Union

What is happening? Is everything working as it is supposed to? Of course not! Even when the button is pressed for a short time, sometimes the program works correctly, and sometimes it skips some positions. Why does this happen?

As you may remember, the processor, in simplified form, performs about 16 million operations per second. Therefore, by pressing the button, the processor will be able to access all the states of our signaling. That said, after letting go of the button there may be a random choice in the sequence.

How to solve this problem? Very simple! It is enough to change the program, so that the change of light does not occur more often than, for example, every second. To do this we can use the already known delay () function.

```
void setup() {
pinMode(10, OUTPUT); //LED vermelho
pinMode(9, OUTPUT); //LED amarelo
pinMode(8, OUTPUT); //LED verde

pinMode(7, INPUT_PULLUP); //Botão

digitalWrite(10, LOW); //Desligar LED
digitalWrite(9, LOW);
digitalWrite(8, LOW);
}

void loop()
{
digitalWrite(10, LOW); //Vermelho
digitalWrite(9, LOW); //Amarelo
digitalWrite(8, HIGH); //Verde

delay(1000); //Parar o programa durante 1 segundo
while (digitalRead(7) == HIGH) {} //Quando o botão é pressionado

digitalWrite(10, LOW); //Vermelho
digitalWrite(9, HIGH); //Amarelo
digitalWrite(8, LOW); //Verde

delay(1000); //Parar o programa durante 1 segundo
while (digitalRead(7) == HIGH) {} //Quando o botão é pressionado

digitalWrite(10, HIGH); //Vermelho
digitalWrite(9, LOW); //Amarelo
digitalWrite(8, LOW); //Verde

delay(1000); //Parar o programa durante 1 segundo
while (digitalRead(7) == HIGH) {} //Quando o botão é pressionado

digitalWrite(10, HIGH); //Vermelho
digitalWrite(9, HIGH); //Amarelo
digitalWrite(8, LOW); //Verde

delay(1000); //Parar o programa durante 1 segundo
while (digitalRead(7) == HIGH) {} //Quando o botão é pressionado
}
```

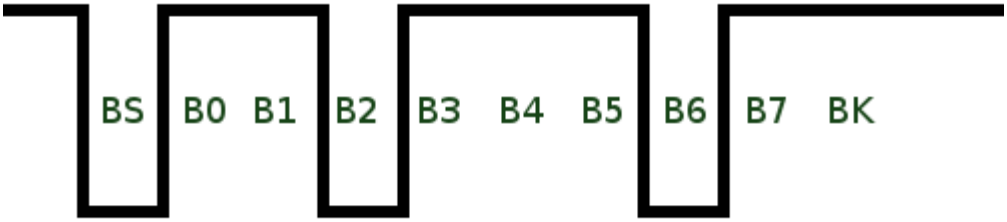
Co-funded by the Erasmus+ Programme of the European Union

Now it should work perfectly! It is important to note that the conditions in the while () function can be combined and much more complex.

UART and Variables

How does the UART work?

Its working principle is based on the serial sending of a sequence of bits, which are transformed into information. A set of data (data frame) is transmitted in the following way:



The transmission starts with the start bit, marked as BS in the figure. It is always a bit that is logical zero. Then, depending on the configuration, there are 7, 8 or 9 data bits (marked B0-B7) that are the information to be sent. The stop bit (indicated as BK) is a logical one bit - it ends the transmission.

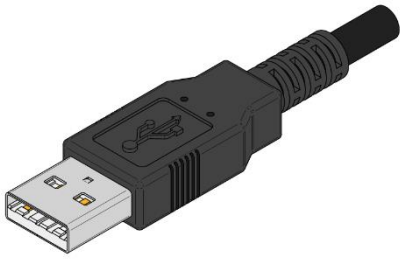
When using the UART in the Arduino, we must take two pins into consideration:

- Tx - sending data (pin 1);
- Rx - receiving data (pin 0).

For transmission to work properly, the same data transfer speed must be set on both systems - known as **baud-rate** or baud-rate. This specifies the number of bits transmitted per second. The most commonly used values are: 9,600 and 112,500.

The computer with which we intend to establish communication must also be equipped with the appropriate interface. Unfortunately, PC manufacturers have stopped inserting the RS-232 serial port, which a few years ago was part of the basic equipment of most computers.

This leaves USB communication. Unfortunately, this is quite a difficult task. Thus, USB-UART converters are usually used, which greatly simplifies the job. The good news is that you don't have to worry about this when using Arduino, the converter is already integrated on the board.



So, all you need to do is connect the Arduino to the computer via the USB cable (the same one that is used in programming).

Co-funded by the Erasmus+ Programme of the European Union

Let's move on to the practical examples. The first two only require connecting the Arduino via USB to the computer. We will only add peripherals to the system later.

Practical Exercise - UART communication

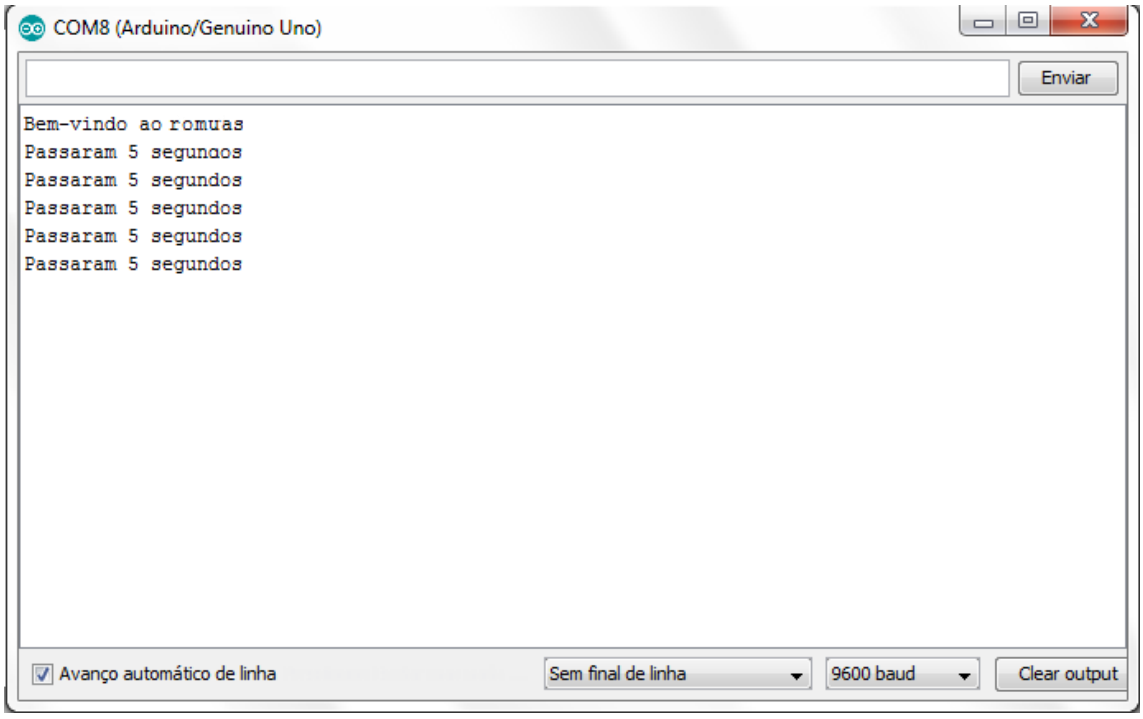
Material required:

- - 1x Arduino UNO and USB cable.

The purpose of the program below is very simple: send a text to the computer:

```
1  void setup(){
2  Serial.begin(9600); //Configuração da velocidade de transmissão
3  Serial.println("Bem-vindo ao ROMUAS!"); //Envio de texto único
4  }
5  void loop() {
6  delay(5000);
7  Serial.println("Passaram 5 segundos"); //Envio de texto em loop
8  }
```

After uploading the above program, apparently nothing happens. To check its operation, you need to select from the Arduino's menu: Tools -> Serial Monitor. After that, a new window will open. Here we can observe what is sent to/from the Arduino through the COM port, which is our UART. Let's see the operation in practice:



Now let's analyse the program. The first thing to be done was to set the baud-rate. The function Serial.begin () is used for this purpose, where in parentheses is the baud-rate. In this case it is 9600 baud/sec. On the other hand, the function Serial.println () is used to send an information (sentence or numbers).

Co-funded by the Erasmus+ Programme of the European Union

The text "Welcome to ROMUAS!" is displayed only once because it is entered in the void setup function and, as you may remember from the previous chapter of the course, instructions entered in this function are performed only once.

Transmission can also be observed in the Arduino's built-in LEDs (Tx and Rx)! These light up when data is being transferred to/from the board.

Practical Exercise - Interaction with the program

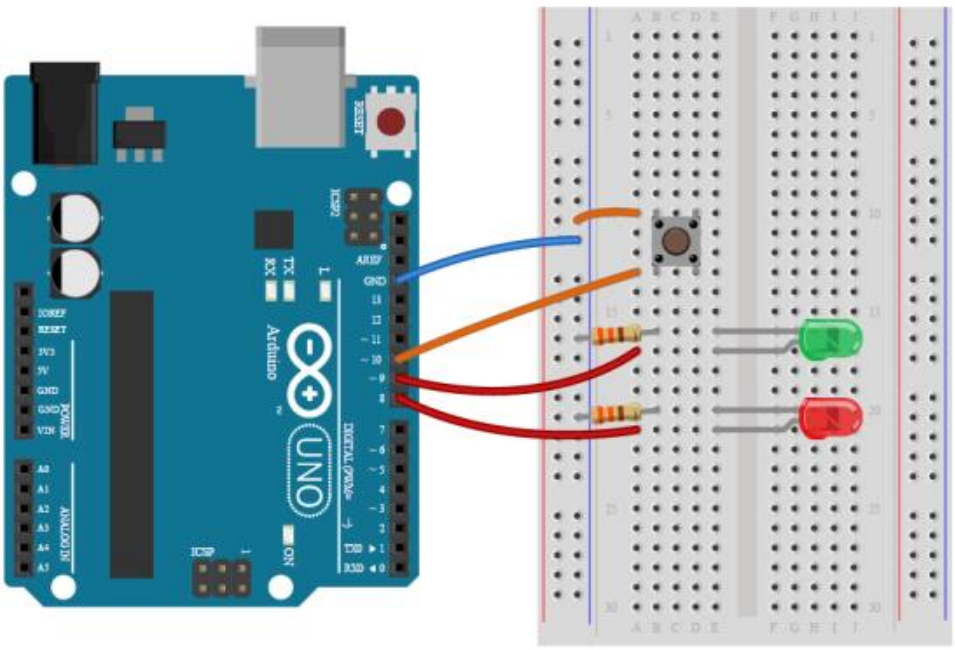
Of course, information doesn't always have to be sent by UART, transmission and reception can also occur once at a chosen time. This is very useful, for example, for diagnosing system operation or signaling different occurrences.

Using the knowledge gained earlier, you can write a sketch that activates an LED when a window is open. Of course, we will not use a window opening sensor, we will simulate using simpler components. A pushbutton will replace the sensor and two LEDs will serve for signaling.

Material required:

- 1x Arduino UNO and USB cable;
- 1x Breadboard;
- 1x Red LED;
- 1x Green LED;
- 1x Pushbutton;
- 2x 330Ω resistors;
- 5x Jumper cables.

Make the connection as follows:



Co-funded by the Erasmus+ Programme of the European Union

When the window is closed (button pressed), the green LED is on. When we open the circuit (stop pressing the button) the red LED lights up and on the serial monitor we will read the message "Attention! Alarm! The window is open!".

```
void setup(){
Serial.begin(9600); //Configuração da velocidade de transmissão

pinMode(8, OUTPUT); //LED vermelho
pinMode(9, OUTPUT); //LED verde
pinMode(10, INPUT_PULLUP); //Botão

digitalWrite(8, LOW); //Desligar LEDs
digitalWrite(9, LOW);
}

void loop() {
if (digitalRead(10) == LOW) { //Se o botão for pressionado
digitalWrite(9, HIGH); //Liga o LED verde
digitalWrite(8, LOW); //Desliga o LED vermelho
} else { //Se o botão não for pressionado
digitalWrite(9, LOW); //Desliga o LED verde
digitalWrite(8, HIGH); //Liga o LED vermelho
Serial.println("Atenção! Alarme! A janela está aberta!");
}
}
```

Let's check how the program is working! Unfortunately, it is not working very well. Alarm information is sent all the time. We prefer it to be sent only once. Do you know how to change this? Sure, by entering the while function!

```
void setup(){
Serial.begin(9600); //Configuração da velocidade de transmissão

pinMode(8, OUTPUT); //LED vermelho
pinMode(9, OUTPUT); //LED verde
pinMode(10, INPUT_PULLUP); //Botão

digitalWrite(8, LOW); //Desligar LEDs
digitalWrite(9, LOW);
}

void loop() {

if (digitalRead(10) == LOW) { //Se o botão for pressionado
digitalWrite(9, HIGH); //Liga o LED verde
digitalWrite(8, LOW); //Desliga o LED vermelho
} else { //Se o botão não for pressionado
digitalWrite(9, LOW); //Desliga o LED verde
digitalWrite(8, HIGH); //Liga o LED vermelho
Serial.println("Atenção! Alarme! A janela está aberta!");

while (digitalRead(10) == HIGH) {
//Criação de um loop vazio para a janela voltar a fechar
delay(25); //Atraso de 25ms dentro do loop para minimizar interferências
}

}
}
```

Co-funded by the Erasmus+ Programme of the European Union

Instruction #define

Over time, our programs will grow considerably. What if we need to change the physical connection of, for example, an LED or a button? Changing the pin number of the entire code would be quite difficult. The #define instruction helps in this regard. It allows you to define a symbol for a certain pin, which will be replaced by its number before compilation, anywhere in the code. For example:

```
#define ledPin 8

void setup() {
pinMode(ledPin, OUTPUT); //Configuração do pino 8 como saída
}

void loop() {
digitalWrite(ledPin, HIGH); //Liga o LED
delay(1000); //Espera 1 segundo
digitalWrite(ledPin, LOW); //Desliga o LED
delay(1000); //Espera 1 segundo
}
```

By putting the line: #define ledPin 8 at the beginning of the code, we make it so that, before compilation, any part of the program that has "ledPin" as pin, is automatically transformed into the number defined for it, namely 8. Of course the pin name can be different, the important thing is to establish a unique name, which will help you write long programs.

IMPORTANT: After the #define instruction no semicolon (;) is used.

Below is the new and improved version of the window opening sensor code:

```
#define LEDvermelho 8
#define LEDverde 9
#define Botão 10

void setup(){
Serial.begin(9600); //Configuração da velocidade de transmissão

pinMode(LEDvermelho, OUTPUT); //LED vermelho como saída
pinMode(LEDverde, OUTPUT); //LED verde como saída
pinMode(Botão, INPUT_PULLUP); //Botão

digitalWrite(LEDvermelho, LOW); //Desligar LEDs
digitalWrite(LEDverde, LOW);
}

void loop() {

if (digitalRead(Botão) == LOW) { //Se o botão for pressionado
digitalWrite(LEDverde, HIGH); //Liga LED verde
digitalWrite(LEDvermelho, LOW); //Desliga LED vermelho
} else { //Se o botão não for pressionado
digitalWrite(LEDverde, LOW); //Desliga LED verde
digitalWrite(LEDvermelho, HIGH); //Liga LED vermelho
Serial.println("Atenção! Alarme! A janela está aberta!");

while (digitalRead(Botão) == HIGH) {
//Criação de um loop vazio para a janela voltar a fechar
delay(25); //Atraso de 25ms dentro do loop para minimizar interferências
}

}
}
```

Co-funded by the Erasmus+ Programme of the European Union

Variables

Before we move on to other programs (including sending information to the Arduino via UART), we have to know what variables are, and how they work.

Variables, in general, are declarations of some kind of information needed by the code. They can be characters, words, or numbers. Most of the time we will come across numeric variables.

When are variables needed? When we want to store a value and perform various types of operations with it. A variable, just like a function, can have a specific type of information, about what kind of data it can store.

Below you can find a list of the most important variable types:

```
boolean logica = false; //Boolean - verdadeiro (true) ou falso (false)

int numero = 30000; //Int - números inteiros no intervalo de -32768 a 32767
long numeroGrande = 2000000; //Long - números inteiros no intervalo de
-2147483648 a 2147483647

float numeroRacional = 6.28; //Float - números racionais que ocupem até 4
bytes de memória

char caractere = 'a'; //Char - caracteres
String frase = "Bem-vindo ao blog ElectroFun!"; //String - sequência de
caracteres
```

NOTE: The maximum values that can be written to a variable depend on the Arduino board used. The above values are appropriate for the Arduino UNO.

At first, we will most often use the following variables:

- **Boolean** - as mentioned, is used to store true or false values. This type of variable is generally used to signal occurrences or control conditions;
- **Int** - is the most common variable for storing integer numbers. It can store information such as the number of keystrokes, how many times a certain situation occurred, or a value given by the distance sensor. And, of course, you can perform mathematical operations on the variables;
- **String** - is a set of characters, that is, in simplified form, we can store a word, phrase, caption, message, etc.

Declaration of Variables

In order to use a variable, you must declare it, that is, tell the compiler about its type and its name. The name of each variable always starts with a letter, never a number, and cannot contain spaces. The declaration of a variable should be done as follows:

```
tipo nome = 0;
```

IMPORTANT: Note that the symbol = is used to assign a variable value, and the symbol == is used to compare the equality between variables and values.

Co-funded by the Erasmus+ Programme of the European Union

It is also important to note that if a variable is placed inside a function, instruction, or subprogram, it will be invisible (we cannot use it) in other functions. Let's see:

```
int variavel = 0; //Variável global - pode ser usada em qualquer parte do
programa

void setup() {
int variavel2 = 0; //Variável local - só pode ser utilizada dentro da função
setup()
}

void loop() {
int variavel3 = 0; //Variável local - só pode ser utilizada dentro da função
loop()
}
```

It is normal that everything is still a little green at the moment, but with time you will learn everything in practice. At first, to make it easier to learn, we will use only global variables. Experienced programmers may not like this, however, we will get to everything, little by little.

There is one more important point to make - naming variables. Remember to give variables names that determine their purpose. For example, instead of:

```
string xx = "Manuel";
```

Prepend a unique name that identifies the variable:

```
string nomePessoa = "Manuel";
```

The problem may arise when the purpose of the variables is more complicated to define. Don't be afraid to create names like:

```
int vMotorEsquerdo = 100; //Velocidade do motor esquerdo
```

The most important thing is to make the code readable and intelligible!

Practical Exercise - Variables

The theory is over, now it's time to practice what you have learned. Let's start with something very simple. To start, let our code write a variable value, which increases with each *loop* cycle.

```
int contador = 0; //Declaração da variável

void setup() {
Serial.begin(9600); //Configuração da velocidade de transmissão
}

void loop() {
Serial.println(contador); //Enviar o valor da variável
contador = contador + 1; //Somar 1 ao valor do contador
delay(100); //Atraso para tornar o programa mais percetível
}
```

Co-funded by the Erasmus+ Programme of the European Union

Obviously, the variable declaration has been placed at the beginning, outside of any function. Thanks to this, we can access the variable anywhere in the program.

First the baud rate is set and initialized, then the loop function () performs 3 actions:

1. Invokes the memory location where we declared a variable called "counter" and sends the value found via UART;
2. Increments +1 on the value received from the counter;
3. Waits 100ms (for a better understanding) and returns to the beginning of the loop.

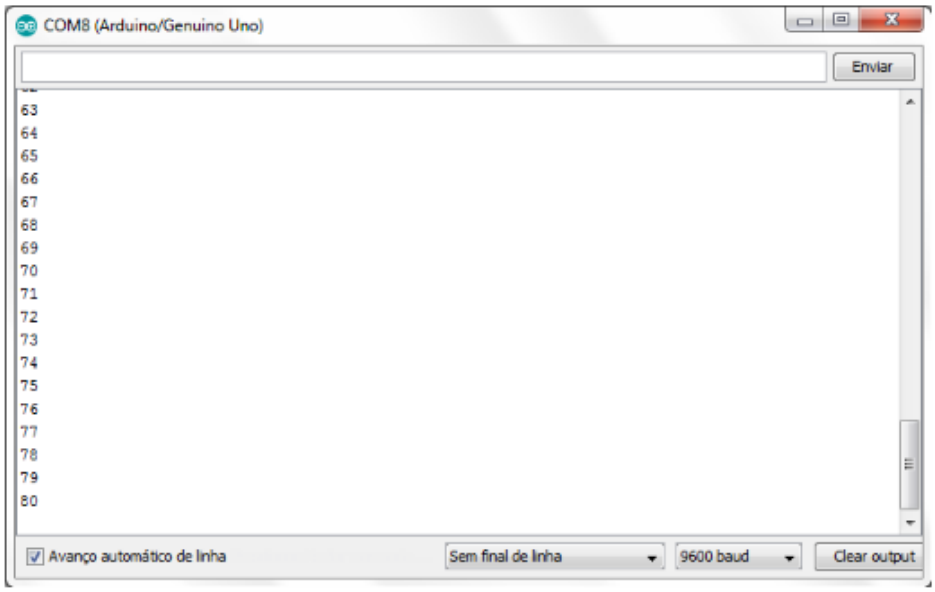
We will now explain point 2, which is the increase of the variable value. The instruction used was the following:

```
1 | contador = contador + 1; //Somar 1 ao valor do contador
```

From a mathematical point of view, where the "=" sign means equality, the above line should not work. However, in programming, the "=" sign means assignment. In practice, the above code should be understood as the following operation:

1. Fetch the value from the counter variable;
2. Add 1 to the received value;
3. Get the result of the operation and assign it to the variable itself.

Download the program to the Arduino and check that everything is working correctly. Of course, to see the results you will have to open the serial monitor.



Co-funded by the
Erasmus+ Programme
of the European Union

Arduino's Bidirectional Transmission

Of course, communication, to be useful, must occur in a bi-directional way. So far, it has been the Arduino sending us information. It is time for us to respond to it!

The goal of the first program is to "hear" our name. When we send it the name, the Arduino should respond with the following message "Hello, Name!", where, obviously, the name will be the one previously sent.

```
String dadosRecebidos = ""; //Conjunto vazio de dados recebidos

void setup() {
Serial.begin(9600); //Configuração da velocidade de transmissão
}

void loop() {
if(Serial.available() > 0) { //Se o Arduino receber dados
dadosRecebidos = Serial.readStringUntil('\n'); // Lê os dados recebidos e
guarda na própria variável
Serial.println("Bem-vindo " + dadosRecebidos + "!"); //Mostrar a mensagem
}
}
```

First, we declare the variable dataReceived, into which the received character set (name) will be copied. Next, as usual, we set the baud rate and start it. Then we introduce a new function: **Serial.available ()**. This sends the number of bytes that have been received and are waiting for support from the Arduino.

In case the data is already available (greater than 0), it is sent to the variable dadosRecebidos. This is done through the .**readStringUntil** (completion) function that copies the data from the buffer until it encounters a completion character (in this case, "\n").

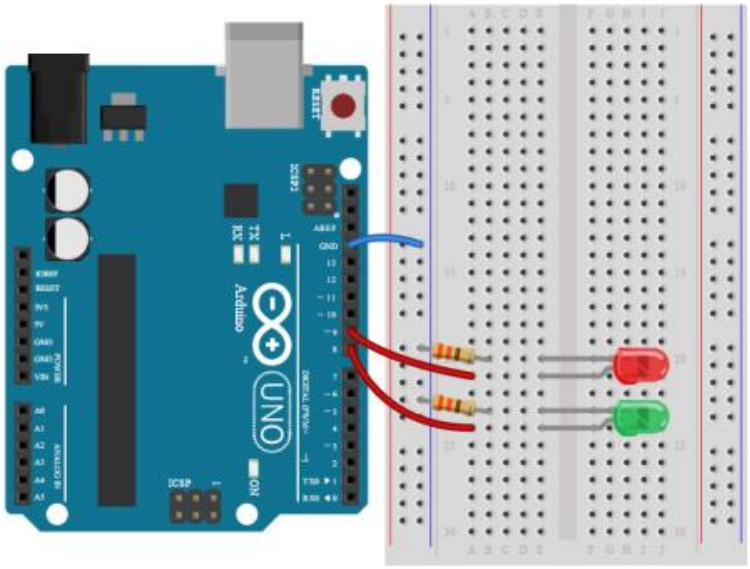
Practical Exercise on Interaction with the System - LED Control via UART

Material required:

- 1x Arduino UNO and USB cable;
- 1x Breadboard;
- 1x Green LED;
- 1x Red LED;
- 2x 330Ω resistors;
- 3x Jumper cables.

In this exercise we will use the ability to send text to the Arduino to control LEDs. To do this, connect two LEDs according to the diagram below (LEDs on pins 8 and 9):

Co-funded by the Erasmus+ Programme of the European Union



The goal of the program is to turn on the green or red LED for 1 second when an appropriate command is sent to the Arduino. The final code is as follows:

```
#define verde 8
#define vermelho 9

String dadosRecebidos = ""; //Conjunto vazio de dados recebidos

void setup() {
Serial.begin(9600); //Configuração da velocidade de transmissão
pinMode(verde, OUTPUT); //Configuração dos LEDs como saídas
pinMode(vermelho, OUTPUT);

digitalWrite(verde, LOW); //Desligar LEDs
digitalWrite(vermelho, LOW);
}

void loop() {
if(Serial.available() > 0) { //Se o Arduino receber dados
dadosRecebidos = Serial.readStringUntil('\n');//Lê os dados recebidos e guarda
na própria variável

if (dadosRecebidos == "verde") { //Se escrever a palavra "verde"
digitalWrite(verde, HIGH); //Liga o LED verde
delay(1000); //Espera 1 segundo
digitalWrite(verde, LOW); //Desliga o LED verde
}

if (dadosRecebidos == "vermelho") { //Se escrever a palavra "vermelho"
digitalWrite(vermelho, HIGH); //Liga o LED vermelho
delay(1000); //Espera 1 segundo
digitalWrite(vermelho, LOW); //Desliga o LED vermelho
}
}
}
```

Let's now take a look at how the program works. Initially, the numbers of the pins with LEDs are set and the variable to which the received data is copied is declared. In the loop it is checked whether the Arduino has received the data. If so, it is ascertained whether this data corresponds to one of the colors. After this, the LED of the indicated color is turned on.

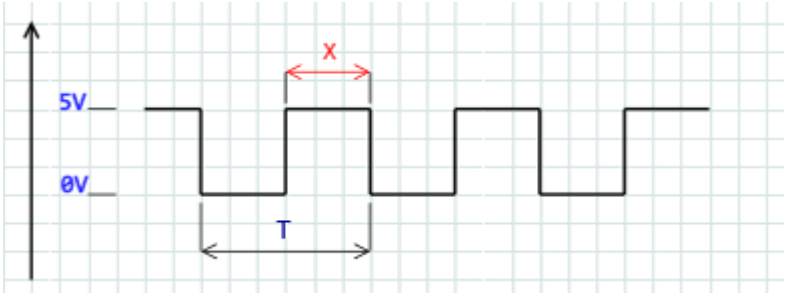
Co-funded by the Erasmus+ Programme of the European Union

What is the PWM signal?

Suppose you connect an LED to the microcontroller and program it to flash in a loop. The LED stays on for one second, the next second it stays off and so on:

```
1  void setup() {
2  pinMode(3, OUTPUT); //Configuração do LED como saída
3  }
4
5  void loop() {
6  digitalWrite(3, HIGH); //Liga o LED
7  delay(1000); //Espera 1 segundo
8  digitalWrite(3, LOW); //Desliga o LED
9  delay(1000); //Espera 1 segundo
10 }
```

If we drew a graph that demontrated the voltage change as a function of time for pin 3, we would get the following waveform:



The value marked as x is the time the LED is on. Conversely, T is the period of time that the LED is off. In turn, its inverse, i.e. 1/T, indicates the frequency. The ratio between the time the LED is on and the time the LED is off is 1:1. In other words, the LED is active during 50% of the program's operation. This is called the ***duty cycle***.

To summarize the signal information above:

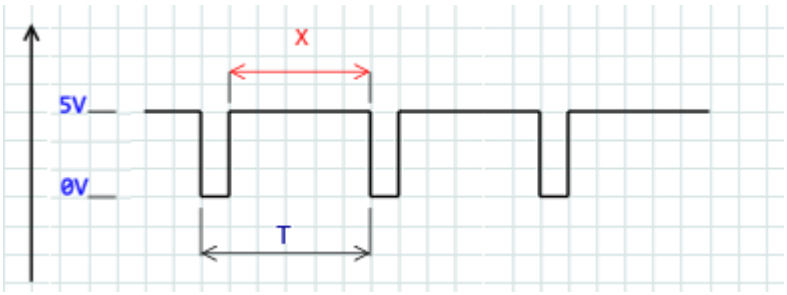
- Amplitude (maximum value): 5V;
- Period (cycle): 2 seconds;
- Frequency: 1/2 = 0.5 Hz;
- Duty cycle: 50%.

Now for another similar exercise. However, with a different duty cycle, yet keeping the period. How to do it? Simply extend the operating time of the LED by reducing the time it is off. For example:

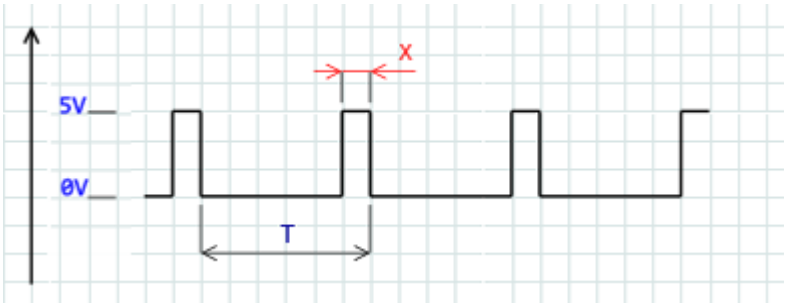
Co-funded by the Erasmus+ Programme of the European Union

```
void setup() {
pinMode(3, OUTPUT); //Configuração do LED como saída
}

void loop() {
digitalWrite(3, HIGH); //Ligar o LED
delay(1667);
digitalWrite(3, LOW); //Desligar o LED
delay(333);
}
```

This time the LED stays on about 5/6 of the time. So the duty cycle is about 83%. Presenting the situation in a graph we get:



Conversely, if we change the delays, the duty cycle of the signal becomes about 17%. Let's look at:



Take a look at the examples above. Which parameter has changed in each example? The answer is easy: the duty cycle. The frequency has remained the same.

Now imagine that the delays inserted in the above codes were much smaller, thanks to which the signal frequency is much higher... Congratulations! You have just understood the principle of PWM. It is a method of modulating a rectangular signal by adjusting the pulse width.

Co-funded by the
Erasmus+ Programme
of the European Union

What is the PWM signal used for?

This signal is used very frequently. With it, you can control the brightness of an LED, the position of a servo and the speed at which a motor runs! As you will see, it has countless applications in robotics, as well as in other DIY projects.

PWM Practical Exercise - LED brightness control

It's time for the first practical example of PWM application. For now, let's create a very simple program, whose goal will be to set the brightness of an LED.

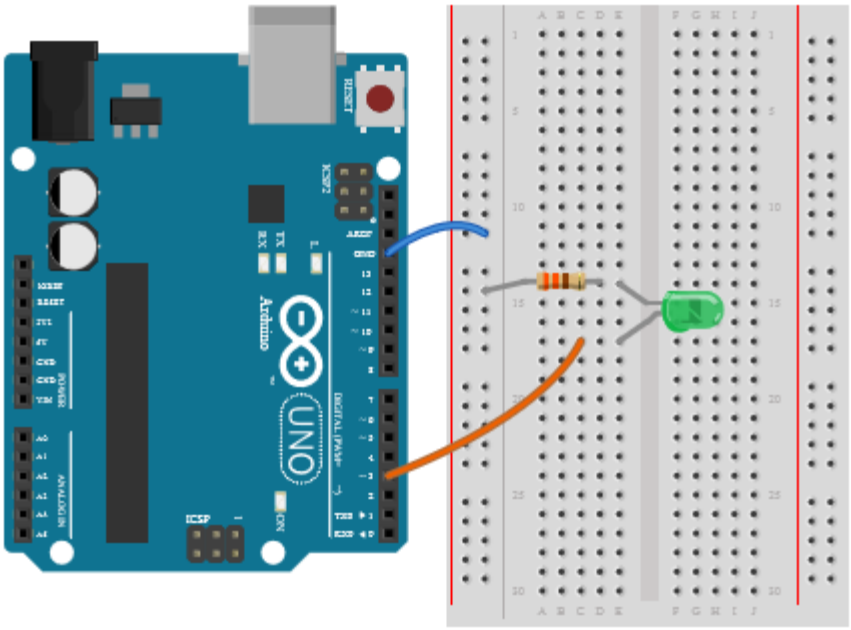
Material needed:

- 1x Arduino UNO and USB cable;
- 1x Breadboard;
- 1x LED;
- 1x 330Ω resistor;
- 2x Jumper cables.

The Arduino is equipped with 6 PWM channels. Each output, on which we can get the PWM signal, has been marked on the board with a tilde "~".

Each PWM channel available in the Arduino UNO is 8 bits long. This means that, the signal we want to receive in its output, can be defined by a number from 0 to 255, where 255 means 100% duty cycle.

To perform the first exercise, you need to connect the LED to pin 3 as follows:



Now let's write the code. The goal is for the LED to light up slowly.

Co-funded by the Erasmus+ Programme of the European Union

```
#define pinoLED 3

int dutyCycle = 0;
int mudanca = 5;

void setup() {
pinMode(pinoLED, OUTPUT);//Configuração do LED como saída
}

void loop() {
analogWrite(pinoLED, dutyCycle); //Gerar um sinal com determinado duty cycle

if (dutyCycle < 255) { //Se o duty cycle for menor do que 100%
dutyCycle = dutyCycle + mudanca; //Aumenta o duty cycle
} else {
dutyCycle = 0; //Se o duty cycle for igual a 100%, volta ao início
}

delay(50); //Pequeno atraso para tornar o efeito visível
}
```

We hope that was clear. We can now discuss the new function that was inserted: analogWrite (pin, duty cycle). Its purpose is to generate the PWM signal at the selected pin with the indicated duty cycle.

The program above intends to periodically increase the duty cycle from zero to the moment when its value is immediately smaller than 255 (100%). When the maximum duty cycle is reached, the LED turns off and the process is repeated.

Servo Motor

It's time to use the servo motor! The one we will use is the SG90, a micro type, one of the smallest available on the market. However, its size does not affect the control mode. Once you understand the principle of operation, you can apply it to larger, more powerful and faster servos.

Co-funded by the
Erasmus+ Programme
of the European Union

What is a servo?

A servomechanism is a motor, gearbox and controller in one device. However, these motors are not designed to perform full rotations. Most of the time, servos have a rotation angle of 0-180°. It is important to know that they know your current position, so you do not need to worry about position errors.

The most important principles of using servos:

1. The position of the shaft should not be turned by hand without necessity. It may damage the delicate plastic gears;
2. You should not feed the servo directly from the power supply used in the rest of the system. Each motor receives a relatively high current, especially at the beginning of the movement. This can disturb the operation of the other devices and, in extreme cases, damage them.

How does the servo work?

How does the servomechanism know which position to turn to? Thanks to the integrated driver. It is he who, on the basis of the supplied PWM signal, controls the motor.

An acceptable standard is to send a signal with a period of 20ms to the servo. The duty cycle is interpreted as the position to which the servo should be moved. The duty cycle of the generated signal should be between 5 and 10%. These values will be converted into two extreme positions on the servo (maximum left and maximum right). Servo connection wires:

- **Red** – power supply;
- **Yellow** or **orange** – signal control;
- **Black** or **brown** – GND.

Depending on the manufacturer, the colours of the threads may vary. However, two will definitely be black/brown and red. The remainder will be the signal wire.

Servo power supply

As mentioned earlier, you should not power the servo directly from the same source that powers the microcontroller. Therefore, due to the fact that the motor consumes a large current, a suitable source must be used to power the system.

Unfortunately, powering from the USB port, as we had done until now, is not enough. Therefore, for the first time, we will power the board with a 9V battery!

Co-funded by the Erasmus+ Programme of the European Union

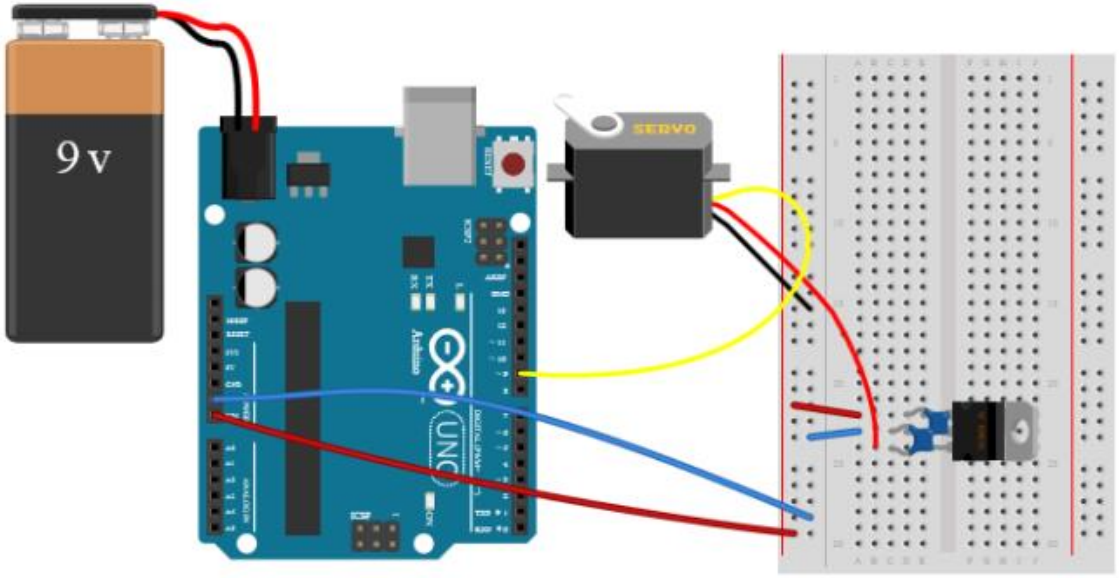
Practical Exercise - Servomechanism

Let's move on to the first program with the servo motor. For this, wire the system according to the assembly diagram below.

Material required:

- 1x Arduino UNO and USB cable;
- 1x Breadboard;
- 1x Servo Motor SG90;
- 1x Voltage Regulator LM7805;
- 1x 9V battery;
- 1x Connector for 9V battery;
- 7x Jumper cables.

Firstly, you need to connect the battery. Secondly, you need to include an LM7805 voltage regulator.



Co-funded by the
Erasmus+ Programme
of the European Union

Now the code that will make the servo move gradually:

```
#include <Servo.h> //Biblioteca responsável pelo servo motor

Servo servomecanismo; //Declaração do servo como servomecanismo
int posicao = 0; //Posição atual do servo de 0-180 graus
int mudanca = 6; //Qual deve ser a posição do servo?

void setup()
{
servomecanismo.attach(9); //Servo ligado ao pino 9
}

void loop()
{
if (posicao < 180) { //Se a posição for inferior a 180 graus
servomecanismo.write(posicao); //Move-se
} else { //Caso contrário, volta ao início
posicao = 0;
}

posicao = posicao + mudanca; //Aumentar a posição atual do servo
delay(200); //Atraso para melhor efeito
}
```

This time we have added a library that will expand the capabilities of the program with our functions. The command used is:

```
#include Servo.h
```

In this case, we add the Servo.h file, which contains additional servo instructions. Thanks to this, we don't need to control the PWM signal ourselves. It is enough to indicate the positions (angle) to which we want the servo to rotate.

To control the servo, we need to declare it:

```
Servo servomecanismo;
```

The **function attach (pin)** - for the Servo object - works in a similar way to **pinMode**. From this instruction, a PWM signal will be generated at the indicated output (in this case 9).

After starting the program, the servo should move smoothly from one extreme position to the other in a loop. The key instruction is:

```
servomecanismo.write(posicao);
```

The position should be an angle between 0 and 180º.

Co-funded by the Erasmus+ Programme of the European Union

Introduction to Displays

Switching on and controlling a graphic or text display can seem very difficult. After all, there are many pixels on the screen and you have to control everything.

Each letter consists of several pixels, arranged in rectangles. The display we are going to use is a 16×2 LCD. This means that it has 2 lines, each with 16 characters. This is a limitation of these small displays. There are others with much more text capacity, where you can display images, etc.

LCD displays are available in various colours, for example green, blue and black. You can use whichever is most convenient for you.

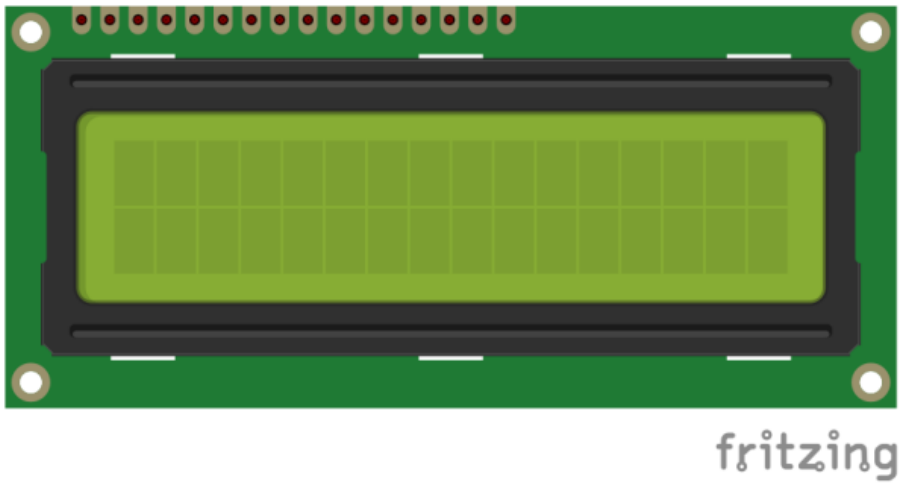
How does the display work?

As mentioned earlier, the display letters consist of pixels. To control each one individually would mean creating code with a multitude of lines. But, of course, this is not necessary, the display works much more easily thanks to the integrated drivers. The best known is the HD44780.

Often, in the description of a certain display, you will find an indication that the display is compatible with the HD44780 driver. When this is the case you know that it will be very simple to operate!

In this case, how can you send a text to the display? It is necessary to connect approximately 12 wires. Of course, only some of them are used for communication, the rest are needed for power supply and other signals.

Most of the time, that display is equipped with a 16-pin connector:



Co-funded by the
Erasmus+ Programme
of the European Union

Let's look at the legend of the pins, from left to right:

1. VSS – ground (GND);
2. VDD – +5V power supply;
3. V0 - contrast adjustment;
4. RS - register selection;
5. R/W - write or read selection;
6. E - signal enable;
7. DB0 - Data;
8. DB1 - Data;
9. DB2 - Data;
10. DB3 - Data;
11. DB4 - Data;
12. DB5 - Data;
13. DB6 - Data;
14. DB7 - Data;
15. LED+ - increasing backlight;
16. LED- - backlight reduction

Pins 1 to 3 are used to power the display, pins 4 to 14 to control the display, and the last two (15 and 16) to regulate the backlight.

Displays compatible with the HD44780 driver can communicate via 4 and 8 bits. In the former mode, 7 connections to the Arduino are required. However, in 8-bit mode, it must perform up to 11 connections.

Connecting the display to the Arduino

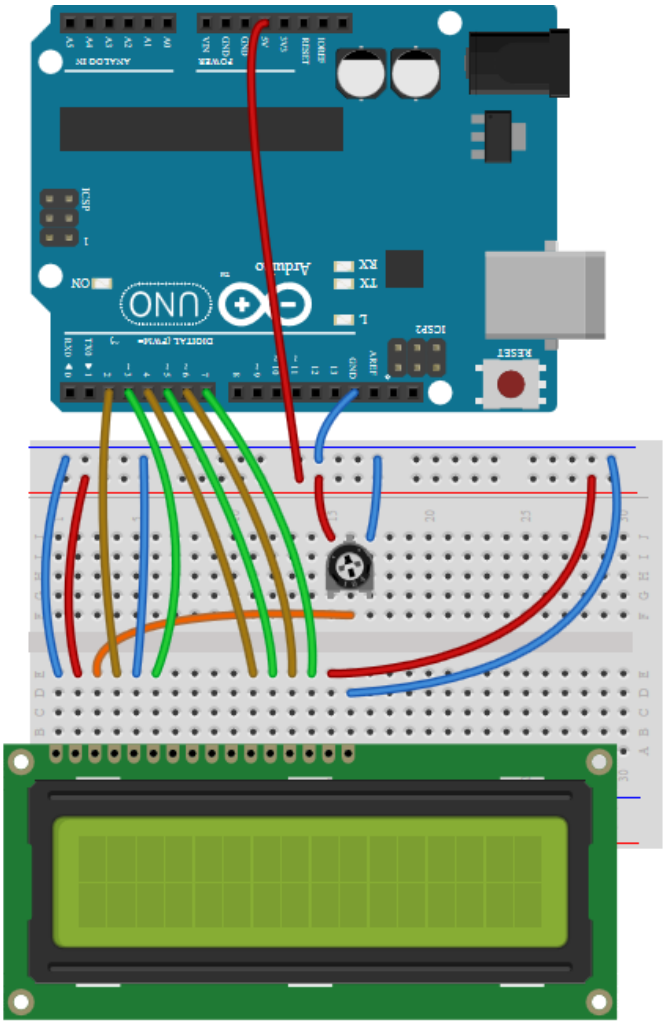
The Arduino has a special library for displaying texts. However, before we move on to programming, we need to make the system connections.

Material needed:

- 1x Arduino UNO and USB cable;
- 1x Breadboard;
- 1x 16×2 LCD display;
- 1x Potentiometer;
- 16x Jumper cables.

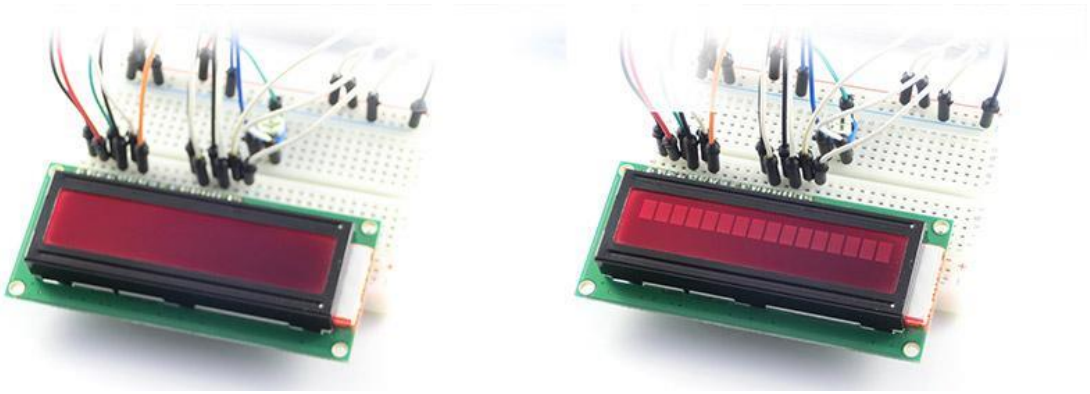
Make the connections taking into account the diagram below:

Co-funded by the Erasmus+ Programme of the European Union



But what is the potentiometer for in this case? It is responsible for adjusting the contrast of the display. As for the backlight, you just connect it to the power supply. However, if you want to adjust the brightness, you can do so by connecting the LED+ pin to a PWM output and program it as you wish. We'll talk about that later.

After connecting the Arduino to the power supply (via the USB cable) the display should light up, as shown below:



When you turn the contrast potentiometer, you may encounter two situations: the normal screen or the top line filled with rectangles. Does this mean some kind of damage? No, it looks good. An

Co-funded by the Erasmus+ Programme of the European Union

efficient mains-connected display should look like this. Therefore, adjust the potentiometer in such a way, that the rectangles are visible.

Practical Exercise - displaying text on the display

Similar to the case of servo motors, there is also a library suitable for the operation of displays. This time, it is titled LiquidCrystal. We will start with an example and then explain the program. Don't download the code for your Arduino just yet!

Material required:

- 1x Arduino UNO and USB cable;
- 1x Breadboard;
- 1x 16×2 LCD display;
- 1x Potentiometer;
- 16x Jumper cables.

```
#include <LiquidCrystal.h> //Biblioteca responsável pelo display
LiquidCrystal lcd(2, 3, 4, 5, 6, 7); //Informações sobre ligação do display

void setup() {
lcd.begin(16, 2); //Declaração do tipo de LCD
lcd.setCursor(0, 0); //Configuração do cursor
lcd.print("Curso Arduino"); //Exibir texto
lcd.setCursor(0, 1); //Configuração do cursor
lcd.print("ROMUAS"); //Exibir texto
}

void loop() {
}
```

The library responsible for the display is in the file: LiquidCrystal.h. To start working on the display, you must declare it. The line used for this is as follows:

```
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
```

Such information indicates that the LCD display was connected to pins 2 to 7 of the Arduino. But, of course, the pin selection is free.

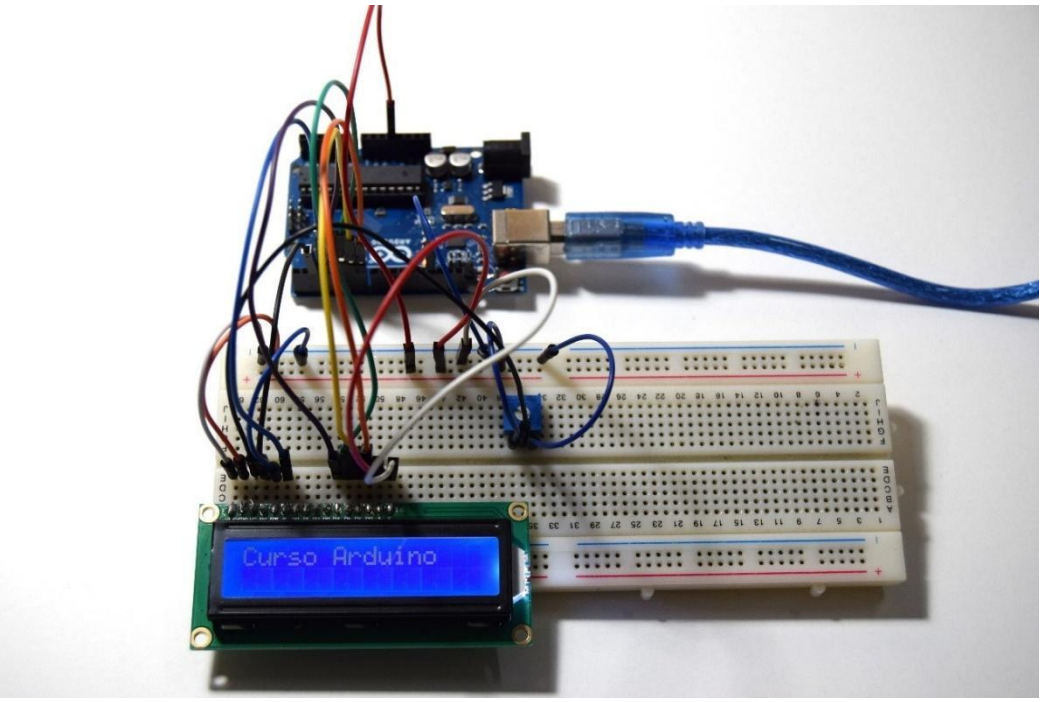
The function lcd.begin (characters, lines) defines the number of characters and lines in which the text will be displayed. In this case, the display allows the display of 16 characters on each of the two lines.

The function lcd.setCursor (position, line) sets the cursor at the given position. For example, a register (0,0) marks the beginning, the first character of the first line. In turn, a register (0,1) indicates the beginning of the second line.

In this case, the most important function is lcd.print ("text"), because it prints the declared text on the screen. When displaying consecutive letters, the cursor moves. That's why the next call of lcd.print starts at the place where you finished the previous text.

Now it's time to upload the Arduino program. The result should be this:

Co-funded by the Erasmus+ Programme of the European Union



What if it doesn't work?

If you can't see the text on the display, check the following points, they solve 99.99% of the problems!

1. Power supply to the display;
2. Connecting the pins on the Arduino;
3. Contrast.

The last point is the easiest to check, and beginners often forget about it! Hopefully after checking the above points everything works perfectly. Now we can discuss the most important functions related to the LCD.

Remove Content from Display

When testing the display of texts on the display, you may find that there are sometimes overlaps. This is because the display does not automatically clear its contents (it has its advantages). However, if you want to delete the previous contents, use the **function lcd.clear ()**.

Co-funded by the
Erasmus+ Programme
of the European Union

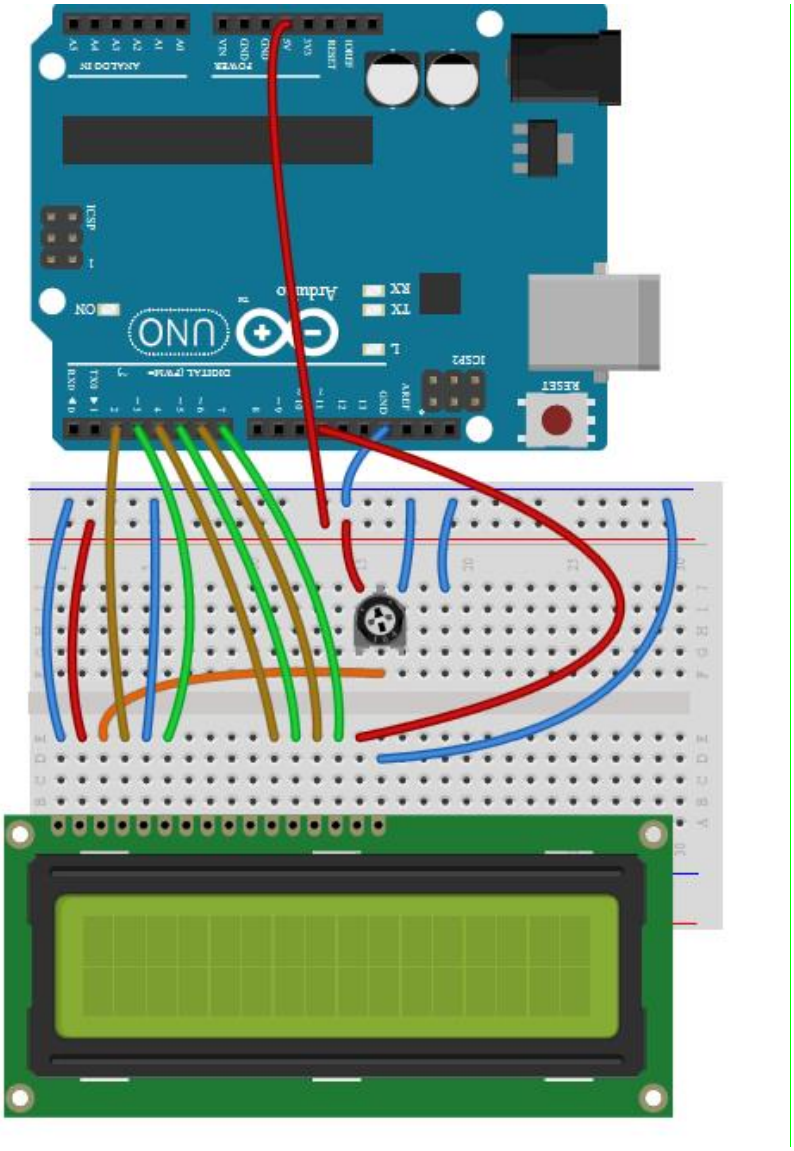
Practical Exercise - Backlight adjustment

As mentioned before, the backlight, as it is an LED, can be dimmed. Let's see how this is done.

Materials required:

- 1x Arduino UNO and USB cable;
- 1x Breadboard;
- 1x 16×2 LCD display;
- 1x Potentiometer;
- 17x Jumper cables.

Make the connections taking into account the following diagram:



Do you remember the possibilities of the PWM signal? Check how the display reacts after you upload the next program.

Co-funded by the
Erasmus+ Programme
of the European Union

```
#include <LiquidCrystal.h> //Biblioteca responsável pelo display
LiquidCrystal lcd(2, 3, 4, 5, 6, 7); //Informações sobre ligação do display

void setup() {
lcd.begin(16, 2); //Declaração do tipo de LCD
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Bem-Vindo ao");
lcd.setCursor(0,1);
lcd.print("Blog ElectroFun");
}
int brilho = 0;
int mudanca = 5;

void loop()
{
analogWrite(11, brilho); //Gerar sinal PWM com determinado duty cycle
brilho = brilho + mudanca; //Mudar o brilho para valor alterado da variável
if (brilho == 0 || brilho == 255) { //Se o duty cycle for 0% ou 100%
mudanca = 0-mudanca; //Alterar brilho na direção oposta
}

delay(30); //Atraso para melhor efeito
}
```

DC Motor Control

Why add motors to Arduino?

Controlling the direction of rotation and the speed of the motor opens the door to many possibilities. With these new capabilities, we can build a simple mobile robot that roams around the house and avoids obstacles. We can also create a vehicle controlled remotely via our mobile phone. The possibilities really are endless!

Which engines are we going to use?

Let's look at the principle of controlling the popular DC motors. These are often used in robotics projects.

Example of a DC motor.

It is still necessary to specify what DC motors are. In general, DC motors are devices that convert electrical energy into mechanical energy. In this specific case, we will talk about motors that consume, on average, less than 1A when powered by 5-9V. These restrictions are conditioned by the motor controller used, but more on this subject later.

Co-funded by the Erasmus+ Programme of the European Union

The principle of motor control is universal: the more powerful the driver, the larger the motor controlled can be.

Why can't we connect the motor directly to the Arduino?

The Arduino, more specifically the integrated microcontroller, controls the signals. The efficiency of each output is relatively small (around 20mA). It is easy to assume that 99.999% of the motors you encounter will need much more current. That said, by connecting a motor to the Arduino you run the risk of irreversibly damaging the board.

Why don't the practical exercises include engines?

As in everything, to start controlling motors, you will have to practice (through the proposed practical exercises). As you can see, in the proposed exercises, there is an Arduino, a battery, a motor controller (L293D), capacitors, but something is missing... no DC motors.

But why? The truth is, including a set of motors would be a waste of your money. Since you are just starting out in this world, you naturally do not know which projects you will develop in the future. Without knowing this, you will not be able to choose the right engines. Why invest in something you will probably never use?

Therefore, in the following exercises, the motors will be replaced by LEDs. You will still be able to observe the change in direction of rotation (through the LED that lights up) and the change in speed (through the brightness of the LED).

For this article you will need the following materials:

- 1x Arduino UNO and USB cable;
- 1x Breadboard;
- 1x 9V battery;
- 1x 9V battery connector;
- 1x L293D chip;
- 1x 1KΩ resistor;
- 1x 10-220uF Ceramic Capacitor;
- 1x 10-220uF Electrolytic Capacitor;
- 2x LEDs;
- Jumper cables.

Introduction to H-Bridges

Intermediate components are needed between the Arduino and the motors. These are often called H-bridges. These drivers can be built from several transistors, or you can simply use an integrated circuit H-bridge. Since you are a beginner, it is best to resort to a ready-made chip.

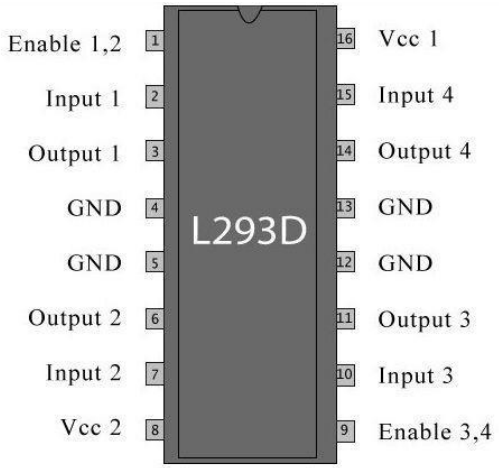
The main purpose of H-bridges is to read and convert the signals sent by the microcontroller into sizes compatible with motor control. For example, the Arduino, whose signals can operate at a maximum of 5V/20mA, after applying the H-bridge, can easily control a motor that requires 12V/1A to operate.

Co-funded by the Erasmus+ Programme of the European Union

Your first H-bridge - L293D

We'll use the L293D chip, which despite being old, is cheap and very common. Its main advantage is that it has a THT assembly, meaning we can connect it directly to the breadboard.

This system has 16 pins. Let's see the legend of each one of them:



One of the key pieces of information about the H-bridge that should be checked is its performance. Specifically, the current that the motor will be able to draw. The L293D has an average current per channel of 0.6A-1.2A. What does this mean? It means that, ideally, motors can draw up to 0.6A, but if for a moment the current increases to 1.2A, nothing bad will happen. As long as it is for a short period of time!

Let's now address each of the pins. First, the **power supply pins:**

- **Ground** (GND) – 4, 5, 12, 13;
- **5V Logic Power Supply** (Arduino) - 16;
- **Power supply for the motors up to 36V** - 8.

Remember that each bridge has a voltage drop. This means that, for example, if you supply a 9V supply, the L293D uses up part of it, and the motors receive 7V at most. This system is old, so it has a large voltage drop. Other, newer bridges (e.g. TB6612) have a lower voltage drop.

Co-funded by the Erasmus+ Programme of the European Union

Motor control pins:

- **Inputs defining the direction of rotation of the 1st motor** – 2, 7;
- **Inputs defining the direction of rotation of the 2nd motor** – 10, 15;
- **Input defining the speed of the 1st engine** – 1;
- **Input defining the speed of the 2nd engine** – 9.

To stop or change the direction of rotation of the motor, you must set the signals according to the following diagram, called **the truth table**:

ENABLE	Input 1	Input 2	Motor
PWM	1	0	Esquerda
PWM	0	1	Direita
PWM	0	0	Parar
PWM	1	1	Parar

Motor pins:

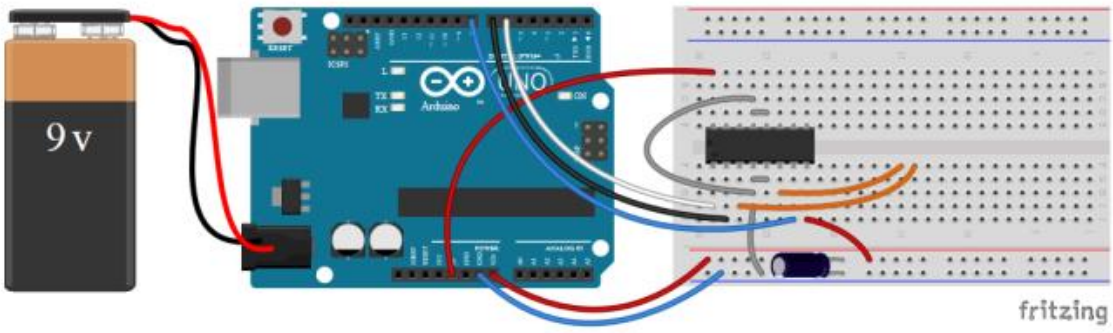
- **1st engine outputs** – 3, 6;
- **2nd engine outputs** – 11, 14.

Practical Exercise - H Bridge

Let's talk about the power supply. We can power the system with one or two voltage sources. If we choose the first option, the Arduino and the motors will be powered from the same source - we don't recommend this configuration, as it can lead to frequent interruptions and resets.

Therefore, it is safer to choose the second option, in which the Arduino will be powered from a separate power supply. In our case, we will use USB power and a 9V battery, from which we will power the motors directly (the Arduino can also be powered this way, but through the built-in stabilizer).

The system should be connected in a similar way to the diagram shown below. Remember that we will connect 9V to the board, not 5V!

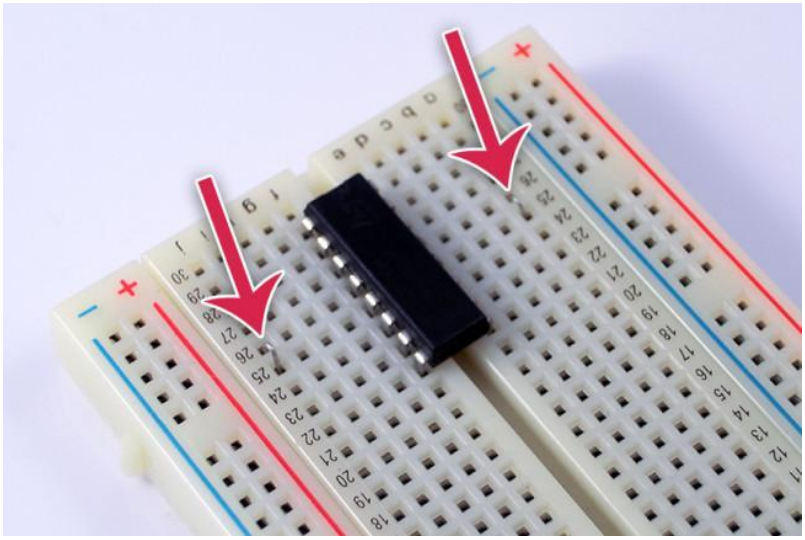


The orange jumper cables will be connected to the "motor".

Co-funded by the Erasmus+ Programme of the European Union

Remember to connect the capacitors, as they are what filter the voltage supplied by the 9V battery.

The system is relatively complicated, so to make the layout more organised, and to save space, we connect the two GND pins on either side of the chip. This can be done as follows, with the legs cut off from the resistors:

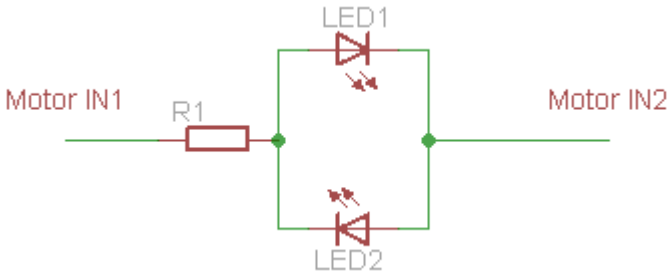


It's time to start the engine. As mentioned earlier, we are going to simulate an engine with LEDs, so you don't have to buy one. But how do we do that? Very simple!

Engine simulation

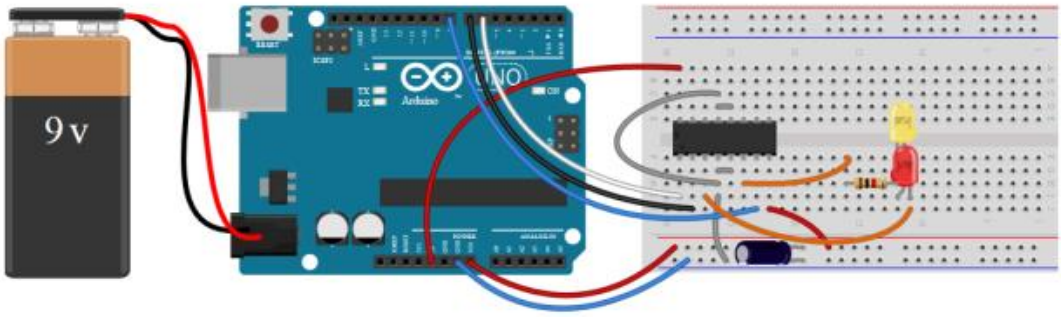
As you know, an LED is a semiconductor element, which lights up when current flows through it in the right direction. Therefore, if we connect two LEDs in parallel so that they are connected in opposite directions, we can check in which direction the current flows - when one LED lights and then the other. In turn, the speed of the motor can be checked by the brightness of the LED.

That said, in place of the motor, we include the following circuit:



Thus, the complete wiring diagram is as follows:

Co-funded by the
Erasmus+ Programme
of the European Union



Programming - Controlling the direction of rotation

We are now going to program, based on the schematic linked above. At the beginning, we will look at controlling the rotation of the motor. We will leave the speed regulation for later. As you can see, the Arduino pins responsible for the motor are the following:

- 6 (PWM) - speed regulation;
- 7, 8 - direction of rotation control.

If we don't want to control the speed of the motor, we need to set pin 6 to high. We could also connect the cable directly to 5V. However, since we already have the connections made, we will use the Arduino:

```
1  void setup() {
2  pinMode(6, OUTPUT); //Sinal PWM do motor para controlo da velocidade
3  digitalWrite(6, HIGH); //Definir permanentemente o estado high no pino 6 -
4  velocidade máxima
5
6  pinMode(7, OUTPUT); //Pino que controla a direção da rotação do motor
7  pinMode(8, OUTPUT);
8  }
9
10 void loop() {
11 //Restante programa
   }
```

If the system has been correctly connected, after uploading the program nothing should happen! Now it's time to add the remaining code to the loop function.

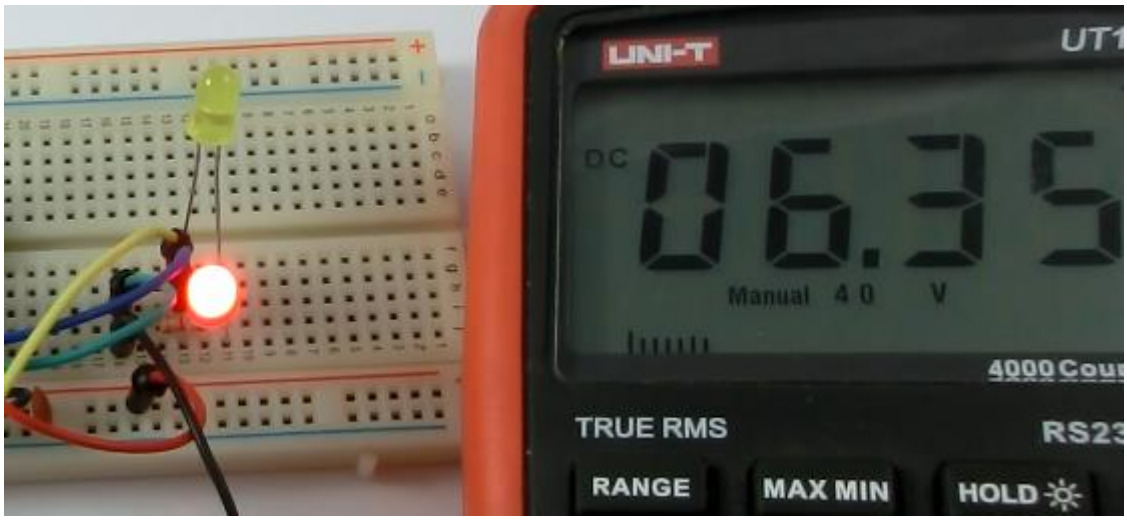
Let's assume that we would like to rotate the motor 3 seconds in one direction and 3 seconds in the opposite direction (maximum speed). For this purpose, we should add a very simple part to the program:

Co-funded by the
Erasmus+ Programme
of the European Union

```
void setup() {
pinMode(6, OUTPUT); //Sinal PWM do motor para controlo da velocidade
digitalWrite(6, HIGH); //Definir permanentemente o estado high no pino 6 -
velocidade máxima

pinMode(7, OUTPUT); //Pino que controla a direção da rotação do motor
pinMode(8, OUTPUT);
}

void loop() {
digitalWrite(7, LOW); //Rodar para a esquerda
digitalWrite(8, HIGH);
delay(3000); //Durante 3 segundos

digitalWrite(7, HIGH); //Rodar para a direita
digitalWrite(8, LOW);
delay(3000); //Durante 3 segundos
}
```

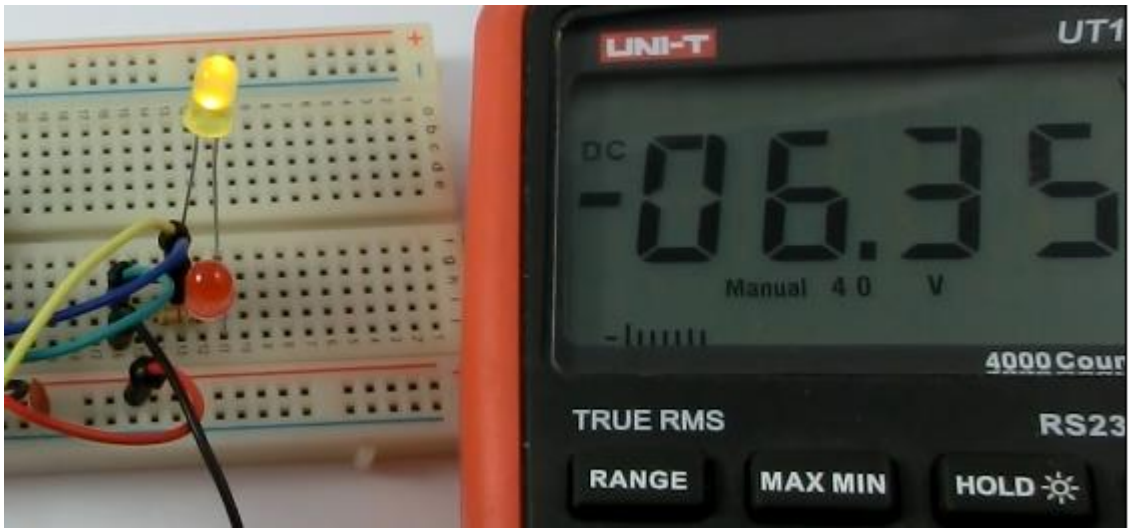
According to the **truth table** presented above, we know that the system works if one of the inputs is in the low "0" state and the other is in the high "1" state. The direction of rotation depends on the input signal.

After uploading the program to the Arduino, the two LEDs should start blinking alternately. Concentrate and understand why this is happening. The LEDs shine alternately because, current flows from output 1 to output 2, and vice versa. The flashing is possible because these have been inserted inversely. To make it easier to understand, look at the following pictures, where, in addition to the LEDs, there is also a voltmeter:



Motor turns to the right

Co-funded by the
Erasmus+ Programme
of the European Union



Motor turns to the left

One time the result is positive and another time the result is negative. In this case, the minus shows that the current flows in the opposite direction to the initial direction.

As we have already managed to control the direction of rotation of the motor, it is time to address the speed.

Smooth motor acceleration

Since we already know the for loop, we can use it to smoothly accelerate the motor. It is enough to change the PWM signal smoothly, for example, every 25 ms.

The code is simple and the variable it is responsible for the duty cycle of the PWM signal:

```
void setup() {
pinMode(6, OUTPUT); //Sinal PWM do motor para controlo da velocidade

pinMode(7, OUTPUT); //Pino que controla a direção da rotação do motor
pinMode(8, OUTPUT);

digitalWrite(7, LOW); //Rodar para a esquerda
digitalWrite(8, HIGH);
}

void loop() {
for (int i = 0; i <= 255; i++) {
analogWrite(6, i); //Aceleração suave do motor
delay(25);
}
}
```

Very often, beginners have problems because the robot does not want to move. Most of the time, it turns out that either the power supply used is too weak, or the PWM signal has too little duty cycle. If you ever have a problem, remember to check these two aspects!

Co-funded by the Erasmus+ Programme of the European Union

3D printing

What is 3D Printing?

3D printing is the name given to a series of techniques that reproduce objects in three dimensions. Also called rapid prototyping, the process allows the printing of models through additive manufacturing. In other words, the printers form three-dimensional models using a technique that superimposes thin layers until the object is ready. Machine parts, decorative objects, jewellery, organs and even food can be printed using this process.

In general, the items are made using some types of plastic as raw material, but it is also possible to use other materials, including metal. 3D printing has revolutionized sectors such as medicine and engineering, besides being very useful for those who want to start an invention. This is because the process shortened the path to reach a test version of any product, called prototype.

Previously, it was necessary to idealize, draw the object and develop a project so that the prototype could be built. With a 3D printer, one only needs to design the item in a specific software and convert the file to a format compatible with the machine to get a prototype. Of course, the use of the software requires some technical knowledge, but the dynamics became much simpler. Depending on the desired item and the 3D printing process, it is also necessary to finish the piece.

How did 3D printing come about?

Although the idea of 3D printing objects sounds futuristic, it is not so recent. In 1984, the first technique used in this process, called stereolithography, was patented at the request of engineer Charles Hull. He created a machine capable of printing special lamps, used in the solidification of resins, as well as plastic parts, from the action of a laser. The initiative was successful, leading Hull to found 3D Systems Corp. to sell his innovative services two years later.

Always focused on innovative technologies, the company has been growing and adding other types of 3D printing in its portfolio, such as Selective Laser Sintering (SLS) and powder-based systems for 3D. Currently, the company has become a reference in the production of customized three-dimensional parts, especially molds used in surgeries and other activities in the health area.

In recognition of his contribution to society, engineer Hull was honoured with the European Inventor Award in 2014. In addition to 3D Systems Corp., it is worth mentioning the collaboration of companies such as Stratasys for the field of rapid prototyping.

Since the 1990s, they have perfected methodologies and developed machines capable of printing different types of artefacts. At first, the printers marketed were more robust, expensive and intended for industrial production.

What is 3D Printing for?

3D printing serves to build various customised objects, in an agile and relatively simple way. After being prototyped, the pieces become real from this process, including details such as cracks, shapes and colours. Because of its versatile character, rapid prototyping serves the most varied purposes. The most common is the printing of a single item, designed for use by one person or a

Co-funded by the Erasmus+ Programme of the European Union

small group. This is the case of stationery items, such as pen holders, and decorative items, such as differentiated frames, lamps and key rings. But the process is gaining popularity in industry, where it helps replace parts, which are produced on demand and at a lower cost than if they were manufactured using traditional dynamics. Depending on the material used, even furniture, tattoos and prosthetics can be printed and successfully perform the functions of the structures that inspired them.

How Does 3D Printing Work?

Just like a regular printer, which releases ink jets in specific quantities and shapes to form images and letters, the 3D printer injects different materials to make up an object. The difference is that, instead of paper, it is on a tray that modern machines print the items. Everything begins with the design of the piece in a 3D modelling software, such as AutoCad and Blender.

There are various types of software, generally geared towards the sector of the product to be printed - construction, stationery, medicine and so on. Then, with the design in hands, the raw material is inserted into the printer, which is turned on and begins to expel the first layer on the tray. In other words, instead of starting from a large piece that will be cut and moulded, the dynamic assembles the articles by joining small parts and layers of material.

The process always starts from the bottom up, printing one layer at a time, and can take from a few minutes to days to complete. Although they follow a similar pattern, there are different types of three-dimensional printing, as detailed below.

Types Of 3D Printer

There are larger and smaller models, aimed at the production of small pieces up to walls and small buildings. They use their own printing techniques, depending on their resources and the raw material they work with.

Know, below, the most popular methodologies.

Fusion Deposition Modeling - FDM

Also called fused filament manufacturing (FFF), this is one of the most commonly used techniques due to its low cost and relative simplicity. This process **uses plastic wires** to print the objects - hence the name of the technique. The filaments are heated, while a head moving along two different axes injects the material into a tray. The plastic is then melted to form the required mould.

The printers that use melt and deposit modelling have **motors that require less power** and cooling to be driven, when compared to lasers used in other modalities of 3D printing. Therefore, machines that print by means of FDM can be installed outside of industrial environments, being employed mainly for **academic purposes** and printing prototypes that will be customised. After all, the technique does not allow much variation in the shape of the object, as it has a simpler finish, suitable for the composition of less sophisticated items.

Stereolithography

In the previous topics, we told you that the pioneering modality was stereolithography, created by Chuck Hull in the 1980s. This technique **employs liquid resin** (acrylic, epoxy or vinyl) to compose

Co-funded by the Erasmus+ Programme of the European Union

each detail of the desired piece. The resin is hardened by an ultraviolet laser beam, which is emitted by the 3D printer to form each layer of the artefact. Finally, the excess liquid resin must be removed and the piece **taken to an oven** to receive the necessary finish.

This method allows the printing of more resistant and complex objects than the FDM technique, however, at a higher time and cost.

Selective Laser Sintering - SLS

This process is similar to stereolithography but uses **powdered raw material** to form the layers of the three-dimensional model. SLS represents great progress for rapid prototyping, as it allows the use of different materials in the printing, generating parts made of polyamides, elastomers, ceramics and metals. The process involves the **use of a robust printer**, which has a printing chamber, in which the powder is injected, levelled and hit by a high-power laser, which heats it and makes the fusion. After finishing the first layer, the printer roller passes over it to cover it with more powder, which will be **heated and fused** until the object is ready. Then, a jet of compressed air and brushes remove the excess powder from the object.

Direct Metal Laser Sintering

As the name suggests, Direct Metal Laser Sintering works by combining **particles made of metal**. As in the SLS, a laser is applied to heat and join the material, resulting in unique products. One of the advantages of this technique is the production of complex shaped items, which normally would not be manufactured in industry.

Polyjet

Polyjet is the methodology most similar to inkjet printing, overlapping the layers with perfection and agility. Using a **type of plastic** (the photopolymer) in liquid form, this process results in a faithful reproduction of the design.

What are the main materials used in 3D Printing?

We list below the main materials for printing three-dimensional models:

- Paper
- Plastics
- Metals
- Rubber
- Glass
- Resins
- Ceramic.

Synthetic plastics are the **most common raw material**, especially for printers that use filaments to make up the layers of objects. Biodegradable polylactic acid, acrylonitrile butadiene styrene and polyamides (such as nylon) are examples of plastics popularly used in 3D printing.

Co-funded by the Erasmus+ Programme of the European Union

What is Tinkercad?

Tinkercad is a collection of software tools, online and free, for those who want to create 3D models in a very simple and intuitive way. Using this tool, you have access to various geometric shapes to build your project. Tinkercad's slogan translates very well its functionality: "from mind to project in minutes".

To use the tool you just need to create a free account at https://www.tinkercad.com/

What are Tinkercad's main tools?

Although Tinkercad is perfect for beginners, this doesn't mean that those who are more experienced with 3D modelling won't like this software. The main idea of the tool is that you create more complex models from the composition of several simple shapes. In addition, the software allows you to add electronic circuits to 3D projects to create objects with light and movement. The final result can even be simulated in the software to check how the components will respond in real life.

Tinkercad provides shapes such as spheres, cylinders, boxes, cones, texts, numbers and connectors that can compose your project. Using these shapes you can **group, duplicate, ungroup, align** or **mirror** until you achieve a satisfactory result for your modeling. And to print the model built in Tinkercad, the good news is that you will be able to export your file in **STL** or OBJ format to be able to perform the pre-press configurations on the slicer of your choice.

Do I need to download TinkerCad?

One of Tinkercad's main advantages is that it is totally online, i.e. you do not need to download it to use it. However, to use it you must register. After creating the account for free, just start modelling!

How to take the first steps in software?

To start, go to the Tinkercad website https://www.tinkercad.com/ . Click on "Sign up" in the top right corner (only if you do not have an account created, otherwise simply login by clicking on "Sign in").

Co-funded by the Erasmus+ Programme of the European Union

On the registration screen you will choose your country and enter your date of birth. Then click on "Next". Enter your e-mail address and your chosen password. Accept the terms of service and privacy and click "Create account".

With the account created the application will automatically open the home screen with the first steps tutorial. If you want to skip this step just click the "X" on the right.

Co-funded by the
Erasmus+ Programme
of the European Union

What are the software's features?

Tinkercad uses the same principle as Lego. In it, you will mainly work with predefined shapes and geometric structures. You can add or subtract pieces to create holes or hollow objects. The first step to start your project is to click on "Create new design".

The work screen will open as in the image below:

1) The rectangle indicated with the number 1 displays the features "Copy" (Ctrl+C), "Paste" (Ctrl+V), "Duplicate" (Ctrl+D), "Delete", "Undo" (Ctrl+Z) and "Redo" (Ctrl+Y).

2) The feature indicated by the red rectangle number 2 is used to move the working plane. The working plane can also be moved by selecting it with the right mouse button. To move in different directions, hold down the selection button.

3) The rectangle number 3 contains the most important functions, which are "Show All" (Ctrl+Shift+H), "Group" (Ctrl+G), "Ungroup" (Ctrl+Shift+G), "Align" (L) and "Flip" (M);

4) The red rectangle indicated with the number 4 refers to the lists of shapes, texts and other resources that the application has to offer.

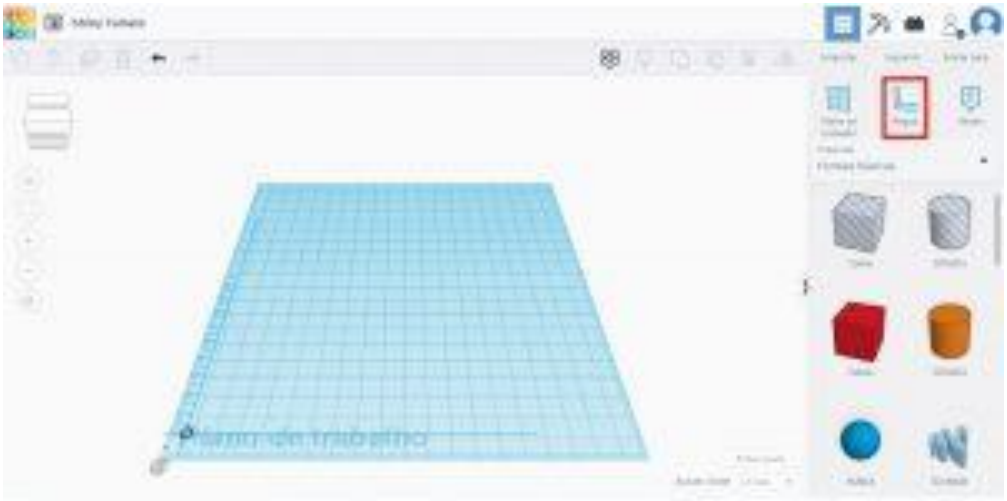
Co-funded by the Erasmus+ Programme of the European Union

How to start modelling with Tinkercad?

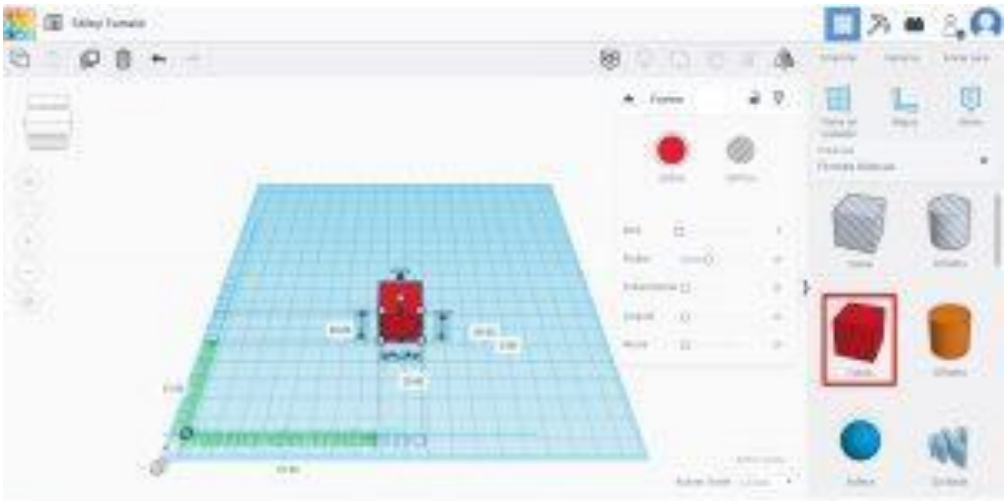
It's time to start your projects in Tinkercad! As an example we are going to show you a luggage identifier template. But you can explore your creativity, since the application allows you to do so.

Step 1: using the "Box" shape

When starting your project, the first (and essential) step is to add the "Ruler" tool located in the upper right corner to your work plan. That way, you'll always have a reference of measurements.



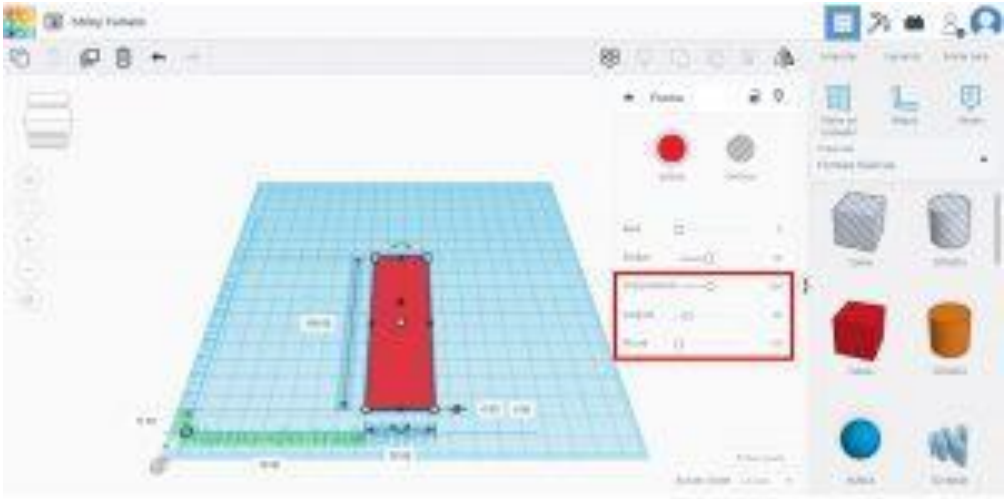
Next, drag the "Box" shape to start modelling.



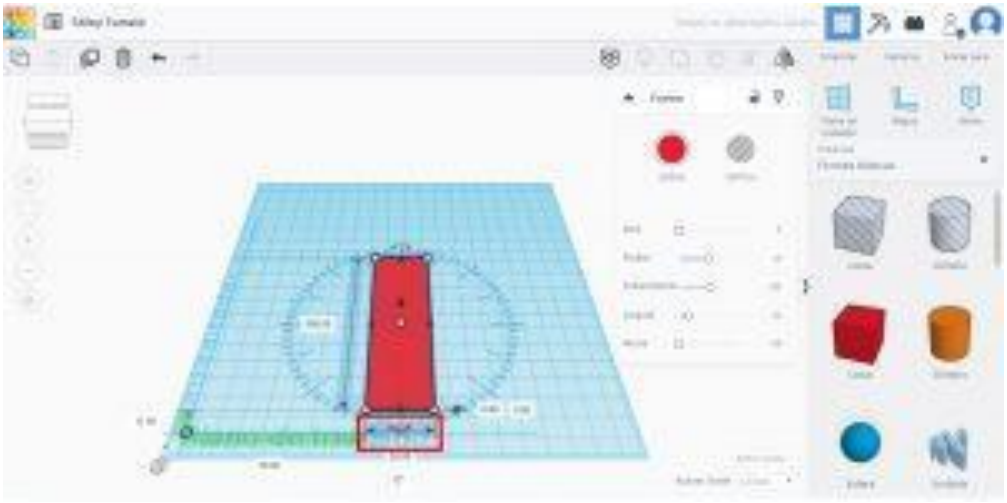
Co-funded by the
Erasmus+ Programme
of the European Union

Click on the object and change it in the shape's own menu. Or, if you prefer, edit the desired dimensions on any of the white squares that will appear around the figure.

In our baggage tag we used 100mm long and 30mm wide. The height is now 0.8 mm.



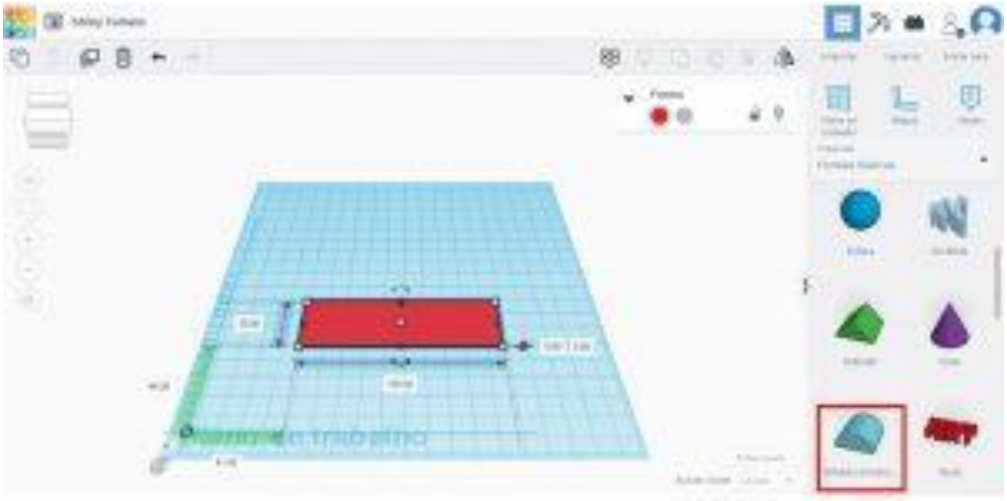
To rotate the piece on the work plane, just click on the picture and move the rotation of the picture.



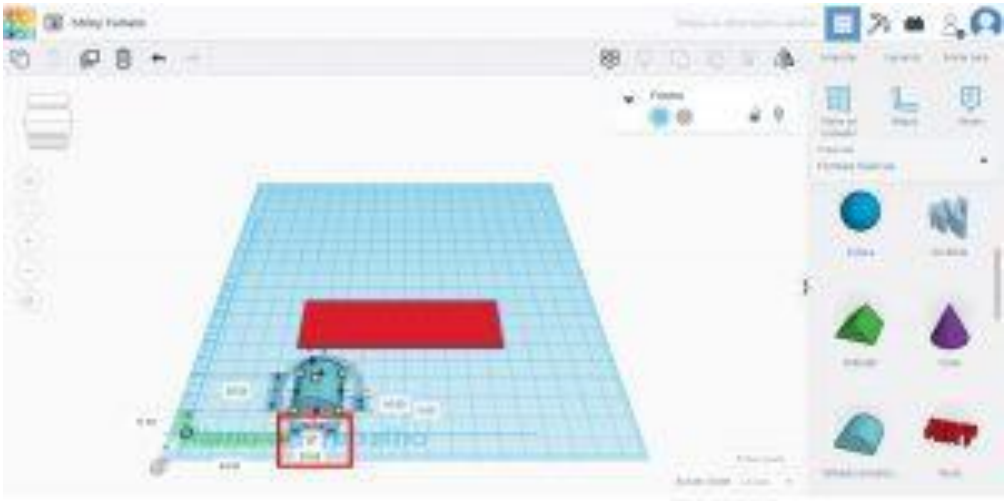
Co-funded by the
Erasmus+ Programme
of the European Union

Step 2: using the "rounded roof" shape

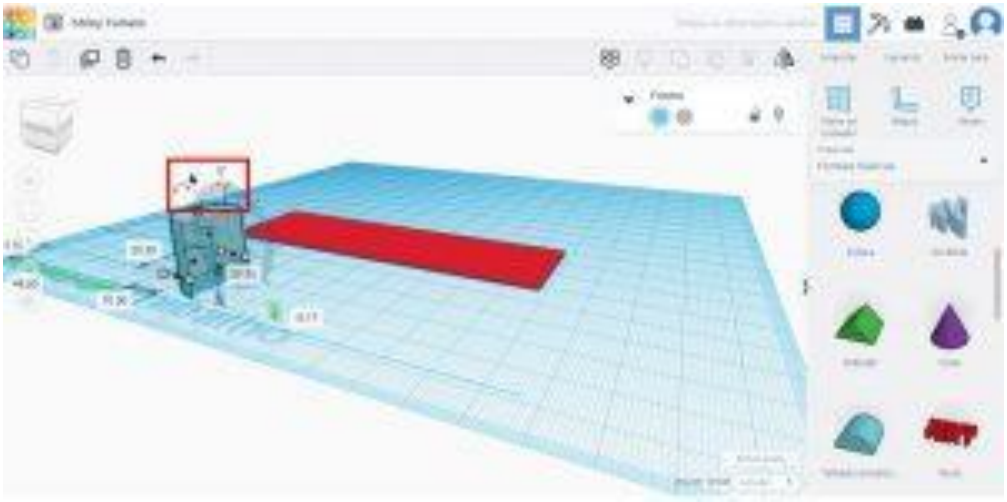
On the side curve of the piece, you can use two different shapes: the cylinder (putting half of it into the rectangular piece), or the rounded roof.



Here we select the rounded roof by rotating it 90° in the XY plane.



To rotate 90° in the Z plane, move the table plane. This makes it easier to see.



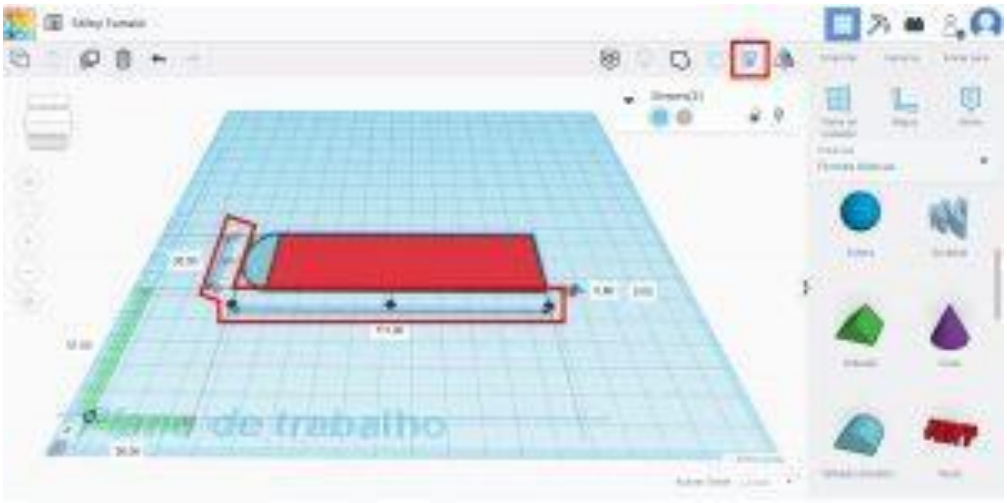
Co-funded by the Erasmus+ Programme of the European Union

Next, change the dimensions of the roof to 10 mm long, 30 mm wide and 0.8 mm high. To do so, click on the white squares as shown in the previous step.

Step 3: aligning the parts

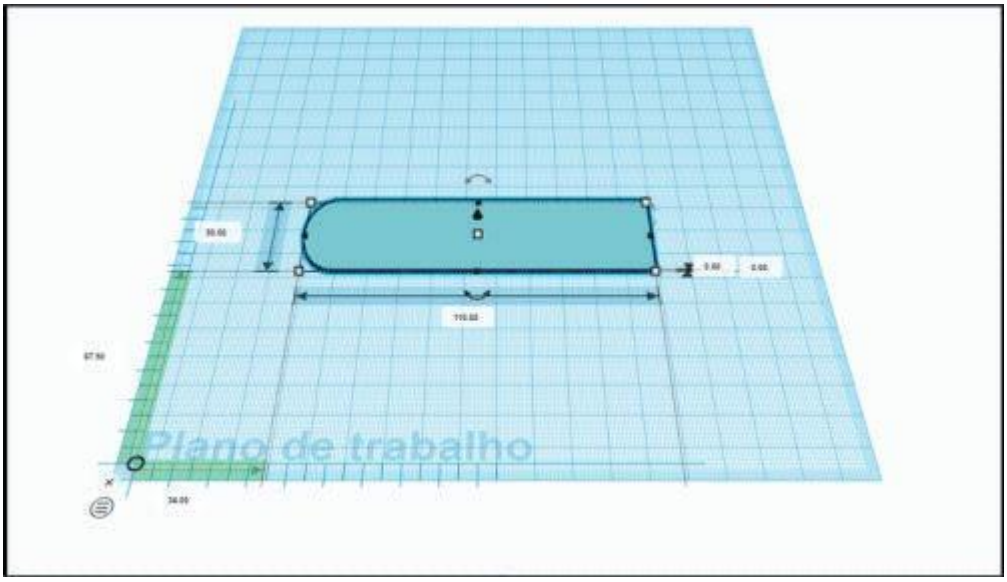
Place the two pieces you have created against each other. To align them, select one of them, hold down "Shift" and click on the other piece. Then, in the top right corner, select the "Align" tool.

Black dots will appear for you to choose the type of alignment. In our case, we left the parts centered.



Step 4: grouping the parts

Now we will group them! To do this, simply keep the parts selected and, in the top right corner, click on the "Group" option. From now on the two pieces will become just one.

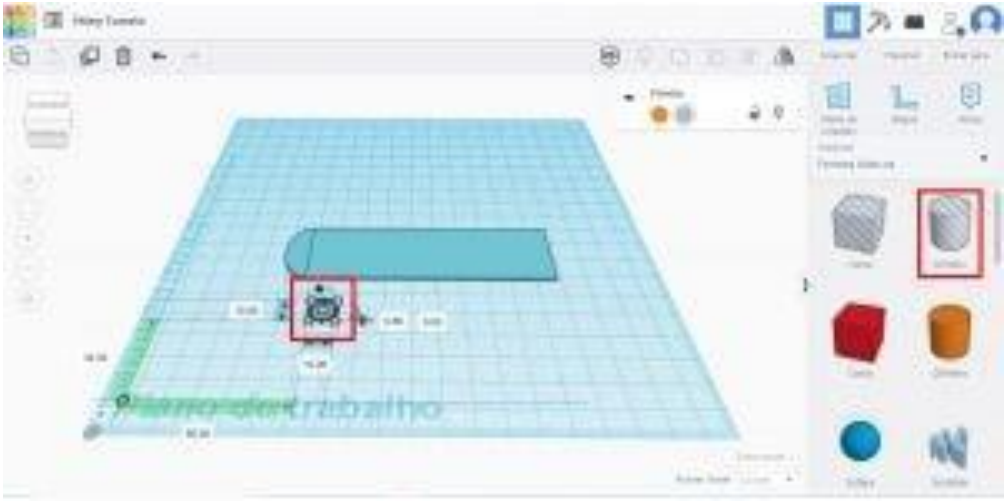


Co-funded by the
Erasmus+ Programme
of the European Union

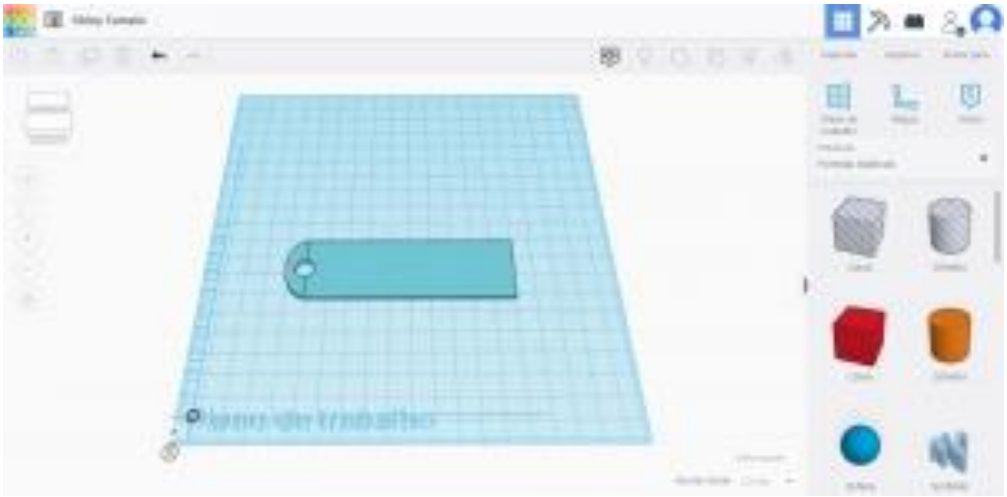
Step 5: Making a hole in the workpiece

There are two ways to drill a cylindrical hole. The first is by selecting the grey cylinder (which even represents a hole). The second alternative is to select the solid cylinder and change it to a hole. In both cases, drag the cylinder over the part, change its dimensions and align the two parts.

In the 3D Lab baggage tag, the dimension used was 10 mm in diameter.



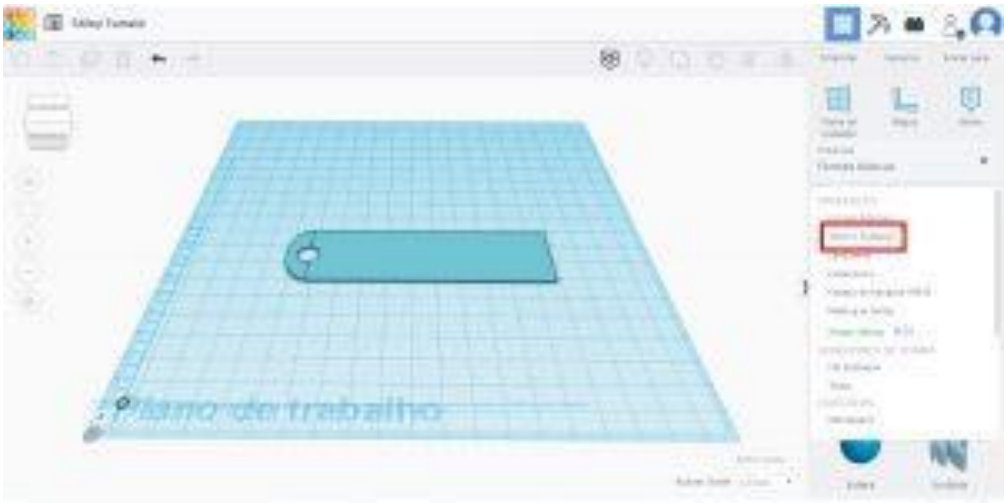
Then select the two shapes and click "Group". Where the cylinder was, a hole will be made in the part as shown below.



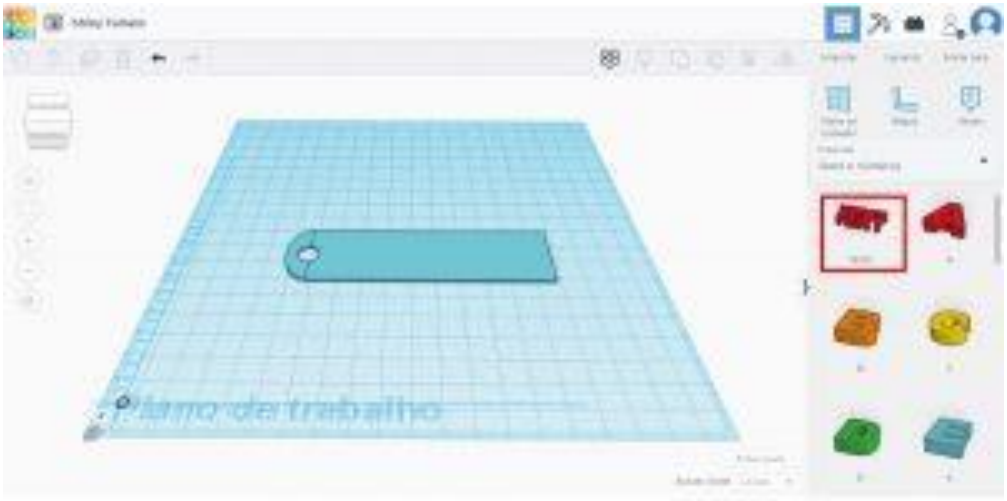
Co-funded by the
Erasmus+ Programme
of the European Union

Step 6: adding the text

On the right-hand side select "Text and Numbers".



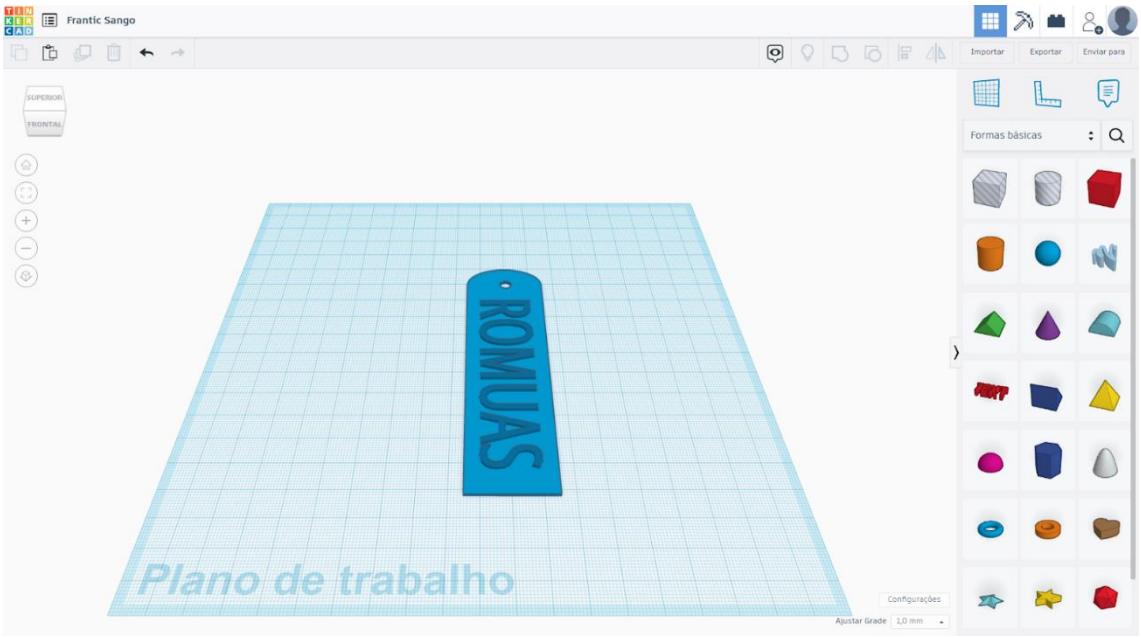
Then drag the "Text" tool into the piece. At this point a side tab called "Shape" will open, and in it you can change the "Text" and the "Font".



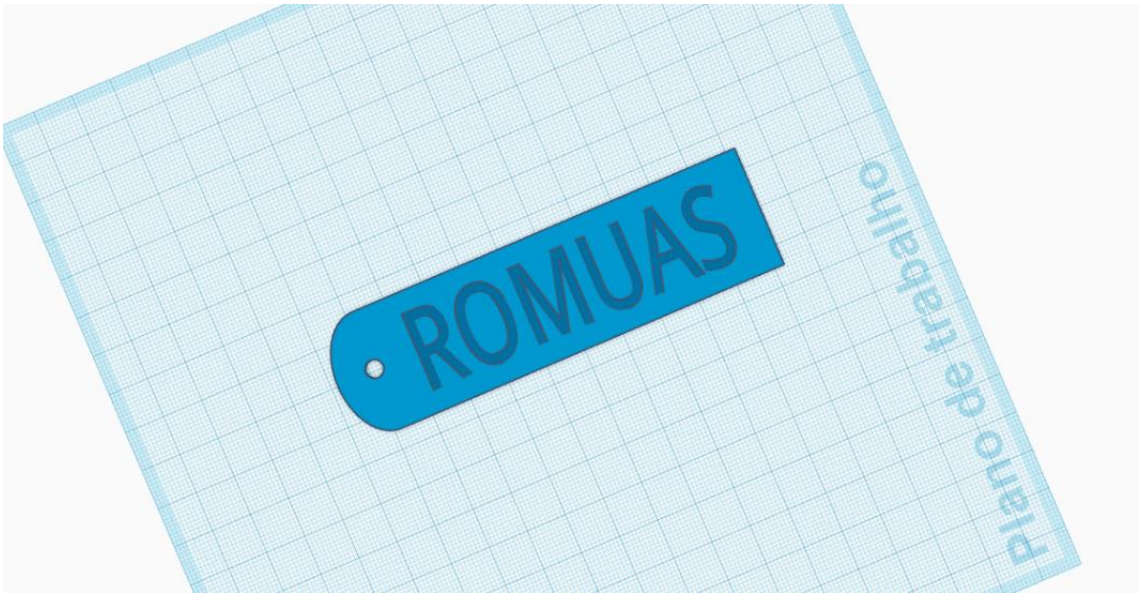
We chose for our example the Sans font. The dimensions of the text were 80 mm long and 20 mm wide.

The height used for the text was 2.0 mm from the working plane since the piece and the text were leaning against the table.

Co-funded by the
Erasmus+ Programme
of the European Union

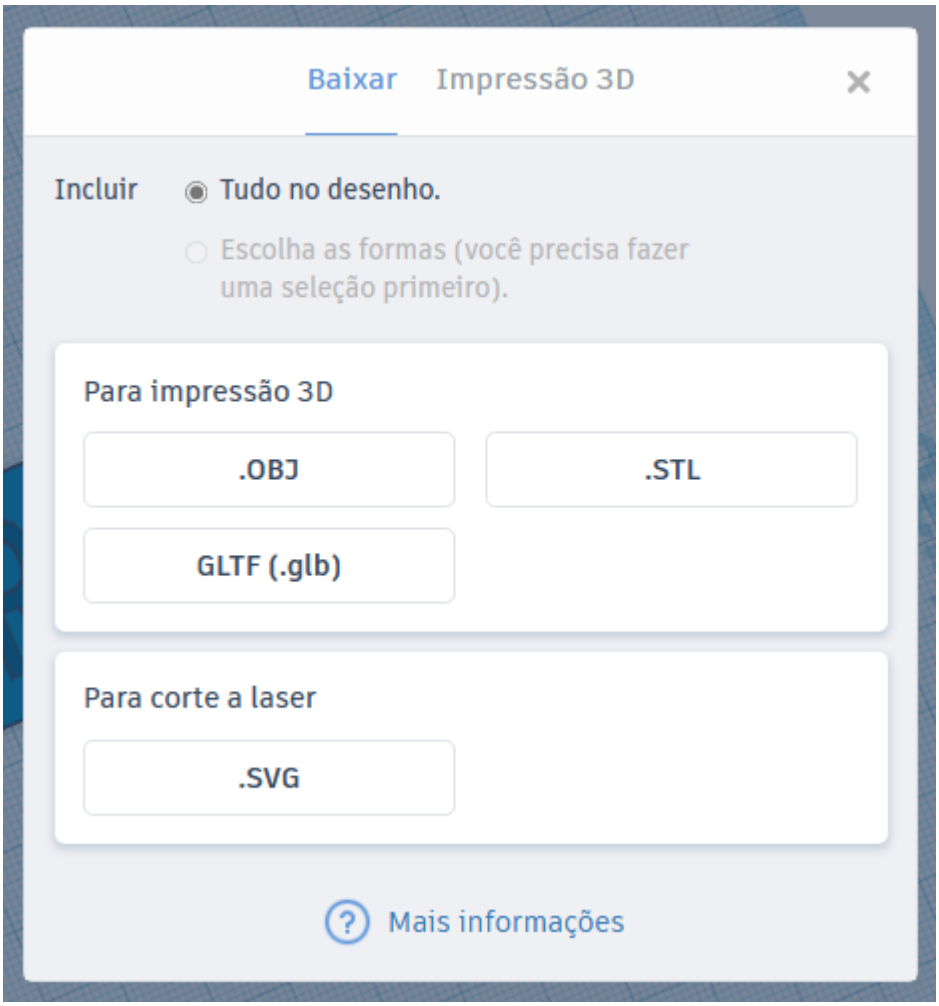


Now group together all the components you have created.



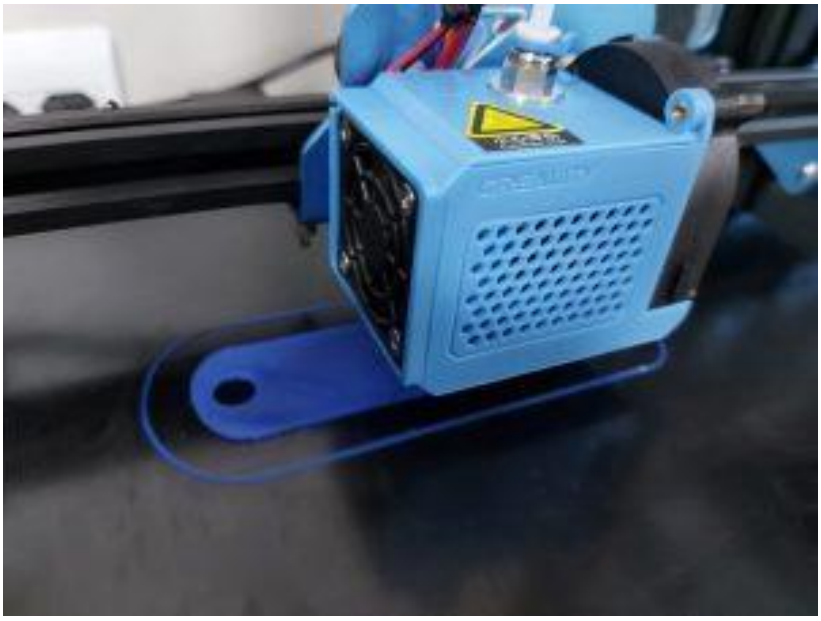
Once this is done your model is finished! To slice and print, simply export as STL through the "Export" tool. It is located in the top right corner.

Co-funded by the
Erasmus+ Programme
of the European Union



Step 7: printing

Finally, once you have your design saved in STL open the file in a slicing software of your choice. Select the print settings and set it to print on your 3D printer.



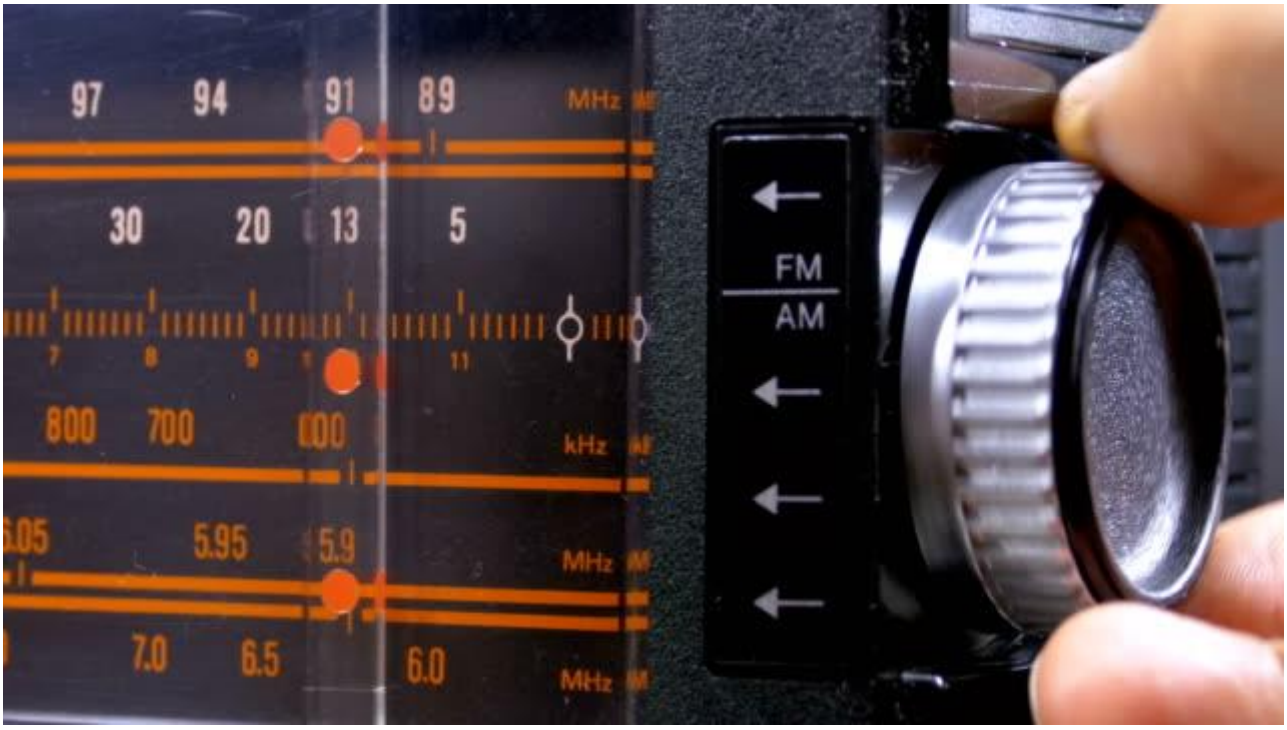
Co-funded by the Erasmus+ Programme of the European Union

II. Analog radio control

Old radio remote control

How it works

Who doesn't have an analog radio receptor? These devices have a wheel to control the dial. You can see the numbers on an analog display called dial.



Example of analog wheel and dial.

We are going to automatize the rotation of the wheel to change the frequency of reception. We will have an app in the mobile phone that will send the desired rotation of memorized dials to the Arduino and it will rotate the wheel to the exact position of this frequency. We will avoid opening or breaking the radio and all our modifications should be external.

Notes:
Every radio receptor is different and you have to adapt the materials and manual to the shape and size of your radio receptor.

Co-funded by the Erasmus+ Programme of the European Union

Materials

We only need an Arduino Uno, a Stepper Motor with an appropriate driver and a Bluetooth controller for Arduino.

Bluetooth controller:

To communicate with the Arduino controller with a mobile phone we need this device. There are many and they are usually cheap.

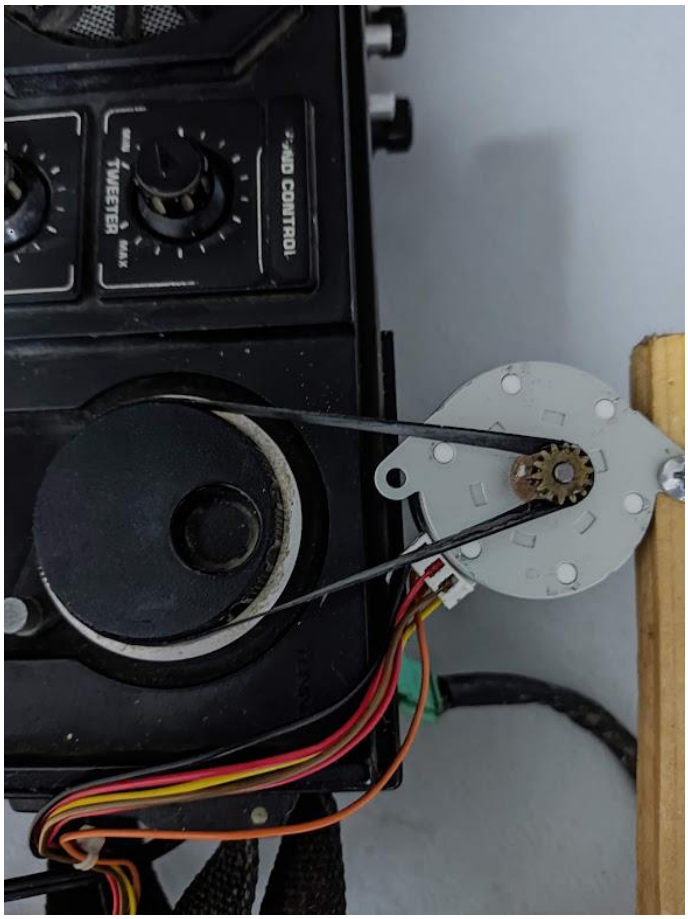
There are lots of steppers and not all work the same way. If we have to buy one, we recommend to search by stepper motor and Arduino to avoid problems and to find tutorials adapted to this microcontroller. Also we recommend buying a stepper motor with the driver in order to not have to adjust voltage and intensity. In the example we will use a reused motor and we will have to use another driver and adjust.

A stepper motor is more difficult to use and slower than a regular motor but it gives us precise control about the exact degrees we want to rotate. And this is what we need.

It is not easy to attach the stepper to the radio. In this example, we are going to show two possible methods. We have this old Sanyo and a reused stepper motor from a printer:

Co-funded by the Erasmus+ Programme of the European Union

As we can manipulate the entire wheel, we decided to use a rubber and use the motor as a pulley:



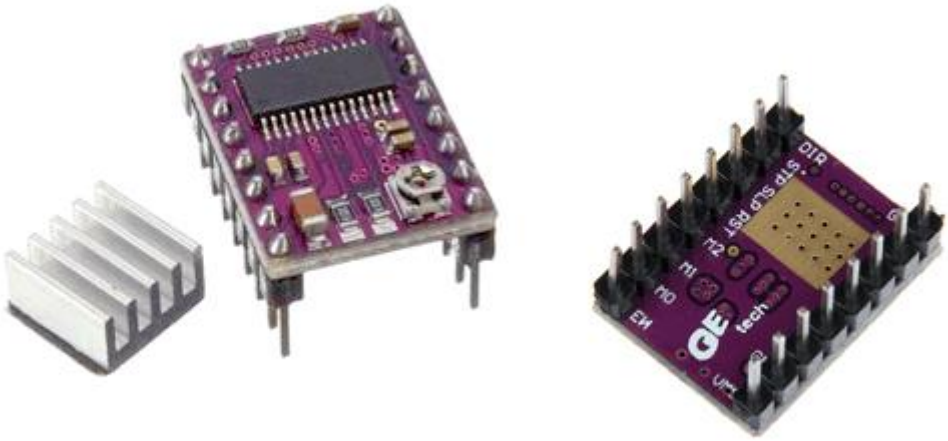
But if it is an inserted wheel or it's impossible to use a pulley, we can glue a Lego or similar wheel or similar and use it by friction. In this case the smaller the wheel the better for more precision and torque:

Co-funded by the
Erasmus+ Programme
of the European Union

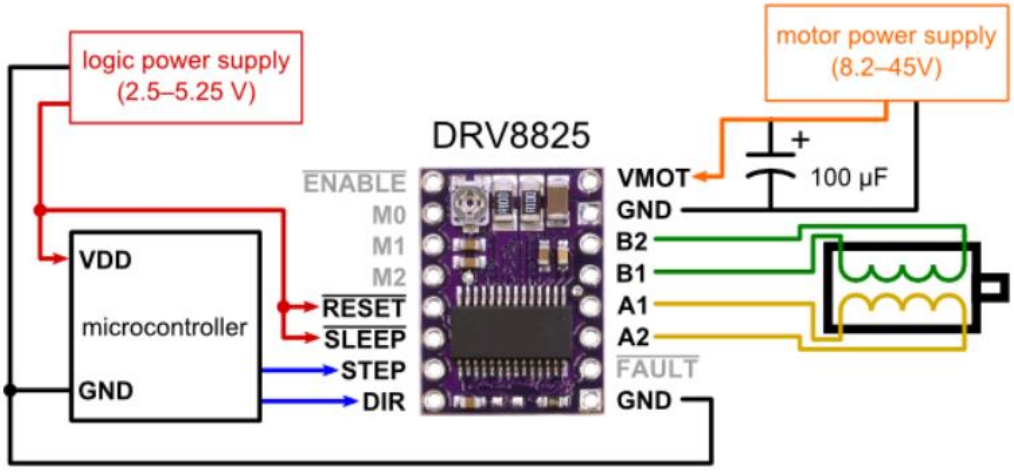
Schema:

If you have a stepper motor with its own driver you don't have to worry about the next part. In the other case, you need a driver:

To control a stepper motor is very difficult with Arduino 'vanilla'. And we have a problem with voltages too. But fortunately we have drivers in the market to control it easier. We can search for A4988 or DRV8825. We recommend the DRV8825 because it is superior in some specifications and the price is not very high. We can find some models because they are used in 3D printers and robots.



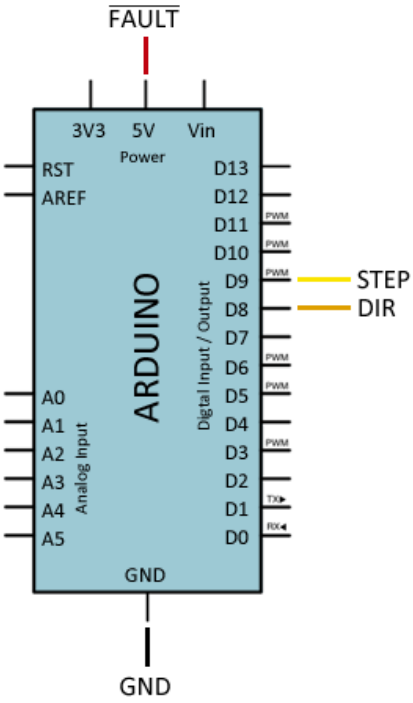
The schema of the drivers is:



Minimal wiring diagram for connecting a microcontroller to a DRV8825 stepper motor driver carrier (full-step mode).

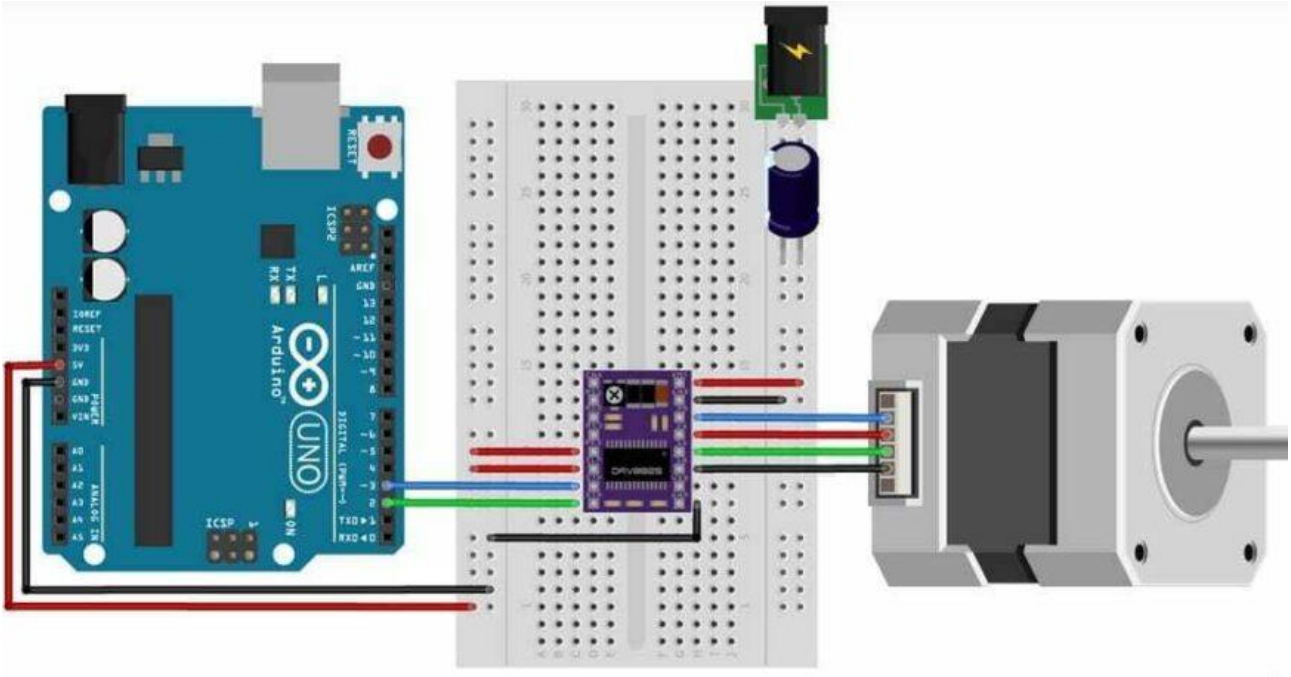
Co-funded by the
Erasmus+ Programme
of the European Union

And in Arduino is:



As you can see, we only need two pins for the steps and direction.

This is the complete schema:



Co-funded by the
Erasmus+ Programme
of the European Union

Assembly process

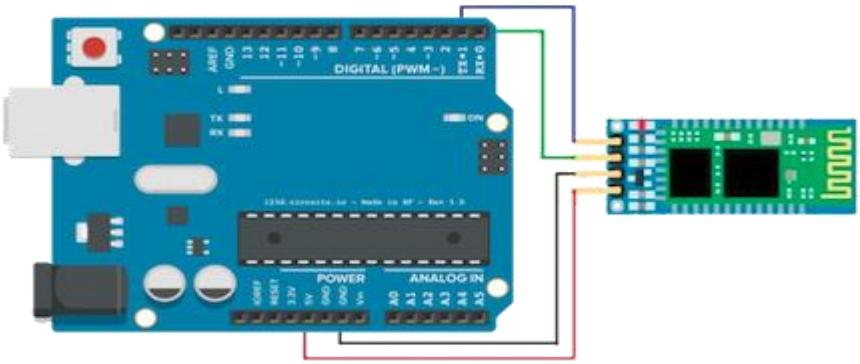
Motors have a nominal voltage and intensity. In the specifications of each motor we can find a formula to calibrate the drivers.

We recommend to read the manual of the driver and the motor in order to adjust correctly the voltage and intensity and not break any of them.

In general the drivers are very strong and difficult to break during work, but we can break it when disconnecting the motor when the controller is powered. Because of this we have to follow these steps:

- Connect the driver without a motor.
- Measure the voltage between GND and the potentiometer.
- Adjust the potentiometer until it reaches the value of the formula.
- Power off the driver.
- Connect the motor and an amperemeter in the middle.
- Adjust to the intensity of the motor.
- Power off
- Mount the motor without an amperemeter.
- Mount to Arduino.

Bluetooth:
It is easy to assemble a bluetooth controller. It uses the serial port of the Arduino and we cannot use serial to communicate with the computer, but we can communicate via bluetooth.



Co-funded by the
Erasmus+ Programme
of the European Union

Code:

This is a simple example to try the motor:

```
const int dirPin = 8;
const int stepPin = 9;

const int steps = 200;
int stepDelay;

void setup() {
  pinMode(dirPin, OUTPUT);
  pinMode(stepPin, OUTPUT);
}

void loop() {
  //Direction and speed
  digitalWrite(dirPin, HIGH);
  stepDelay = 250;
  // 200 steps is a full turn
  for (int x = 0; x < steps * 1; x++) {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(stepDelay);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(stepDelay);
  }
  delay(1000);
}
```

For the Arduino control, here we have a minimum example:

```
const int led = 13;
void setup(){
 Serial.begin(9600);
 pinMode(led, OUTPUT);
}
void loop(){
 if (Serial.available()>0){
   char option = Serial.read();
   if (option >= '1' && option <= '9')
   {
     option -= '0';
    for(int i=0;i<option;i++){
       digitalWrite(led, HIGH);
       delay(100);
       digitalWrite(led, LOW);
       delay(200);
       Serial.println(i);
    }
   }
 }
}
```

We can make an initial program that counts how many steps we need to go through all the dial. This is the calibration of the Arduino analog radio system. We put by hand the dial in the first frequency and with an app for making Bluetooth remote controller, we send a step quantity and the Arduino returns us the count of steps:

Co-funded by the
Erasmus+ Programme
of the European Union

```
const int dirPin = 8;
const int stepPin = 9;

const int steps = 200;
int stepDelay = 250;
int countSteps = 0;

void setup() {
  pinMode(dirPin, OUTPUT);
  pinMode(stepPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {

if (Serial.available()>0){
   char option = Serial.read();
   if (option >= '1' && option <= '9')
   {
     option -= '0';
      digitalWrite(dirPin, HIGH);
      digitalWrite(stepPin, HIGH);
      delayMicroseconds(stepDelay);
      digitalWrite(stepPin, LOW);
      countSteps += option;
   }
    Serial.println(countSteps);
 }
}
```

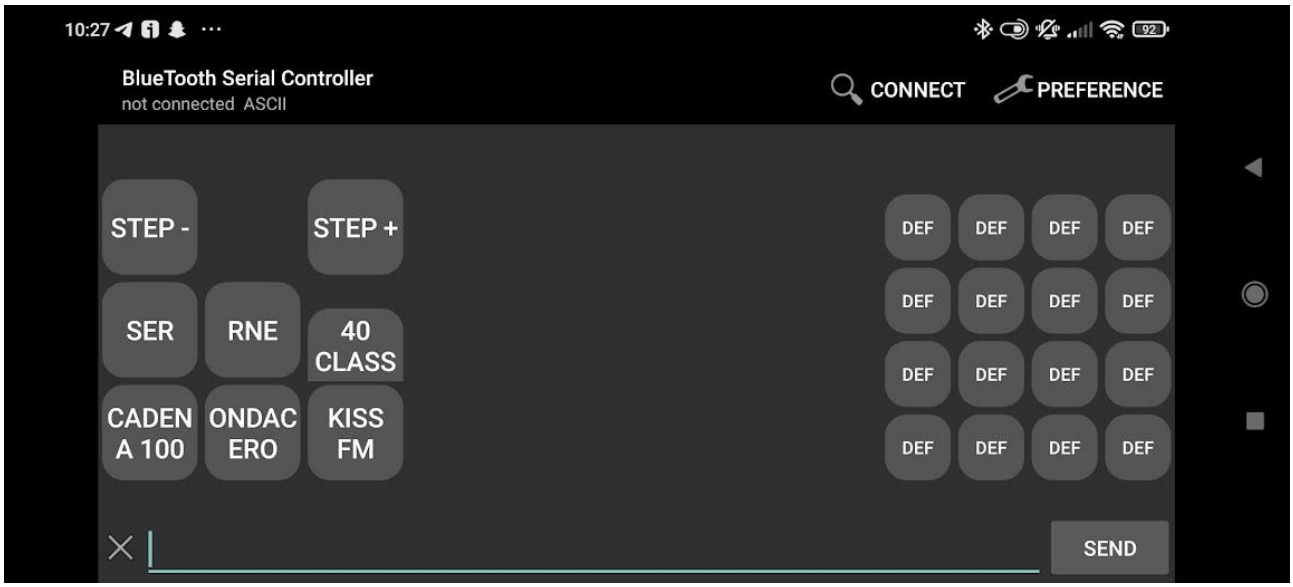
Imagine we need 1000 steps to complete the dial of the example radio. We can hardcode this number in a new code as the maximum of steps. The new code has to know the current position of the wheel and will receive the number of steps since the beginning for a certain station. We can make a button for our favorite stations in an app for controlling Bluetooth and put the desired station with one button. Here we have a simplified code:

```
const int dirPin = 8;
const int stepPin = 9;

const int steps = 200;
int stepDelay = 250;
int countSteps = 0;

void setup() {
  pinMode(dirPin, OUTPUT);
  pinMode(stepPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
if (Serial.available()>0){
   int dial = Serial.parseInt();
   int stepsDifference = dial - countSteps;
   if(stepsDifference < 0){
     digitalWrite(dirPin, HIGH);
   }
   else {
     digitalWrite(dirPin, LOW);
   }
   for(i=0;i<abs(stepsDifference);i++){
      digitalWrite(stepPin, HIGH);
      delayMicroseconds(stepDelay);
```

Co-funded by the
Erasmus+ Programme
of the European Union

```
    digitalWrite(stepPin, LOW);
    countSteps += option;
    Serial.println(countSteps);
  }
 }
}
```

And this is an example of the app Bluetooth Serial Controller for Android:



You only have to assign the number of steps since the beginning to each dial.

III. Water pump and valve control

Drop irrigation for garden and fields

How it works

The starting point is a manual opening valve and an electric water pump to provide the necessary pressure. This situation is inefficient because you have to open the valve, start the pump, wait and observe if the field is irrigated enough and then, stop the pump and close the valve. It's difficult to not use too much or less water and you need to stay during the irrigation. An improvement is to install a programmer to irrigate some minutes every day. The problem with the programmer is the weather, because it will irrigate even on a rainy day and may not be sufficient water on a very hot summer day. There is another problem: water pumps often provide more pressure and flow than necessary. They are oversized. It causes good irrigation because drip irrigation systems can self-compensate and flow the same liters per minute. But the water pump is using more energy than necessary.

Our system has field moisture sensors and uses it to decide when to start irrigation. In addition, it has a pressure sensor and a motor driver to control the speed of the water pump to reach the minimum required pressure.

Co-funded by the Erasmus+ Programme of the European Union

Notes:

This document describes a complete scenario. If it's too difficult or long, you can choose the pump with a pressure sensor or the valve with a moisture sensor.

Materials

In lab tests need a realistic set to recreate a real scenario. We need a pot with plants placed higher, a self-compensated dripper and a hosepipe. For the electronic part, we need a water pump, a motor driver, moisture sensor and an arduino. Now we are going to be more specific:

Dripper:

Katif-like drippers are the easiest for compact installation. You can choose Katif for 1, 2 or 4 liters per hour. We recommend less water per hour for the lab set. These are self-compensate drippers and always drops the same water regardless of the pressure. We recommend using a long hose pipe because it will last more time to fill and will be more realistic.

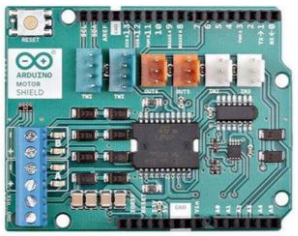
Water Pump:

There are many 12v water pumps for camping or aquariums. Arduino works with 3v and 5v and for electronic protection it's not recommended to move a motor directly from Arduino. For this reason, we have to use a motor driver. It will transform our 5v PWM signal into a 12v signal for different speeds in the motor.

Driver:

The official Arduino Motor Shield is the one we recommend. It is not very expensive, compatible with Arduino Uno and you can find a lot of documentation about it.

A shield is a board that can be placed on top of the Arduino and whose connectors adapt perfectly in the Arduino Uno. The shield uses some of the pins of the Arduino and exposes all the others.



Co-funded by the Erasmus+ Programme of the European Union

Moisture sensor:
This sensor acts by measuring the conductivity of the soil. There are many different sensors and we have to calibrate every time, because the sensibility of each could be different. In this example we will use the Sparkfun model that needs two wires for power and one for analog signal.

Pressure sensor:
This pressure sensor has to be installed in the pipe after the pump. It has tree wires: two for power and one for signal.

Schema:

First, we have to explain the pins used for Arduino Motor shield:

The technical specifications and how to mount it can be consulted here: https://store.arduino.cc/arduino-motor-shield-rev3

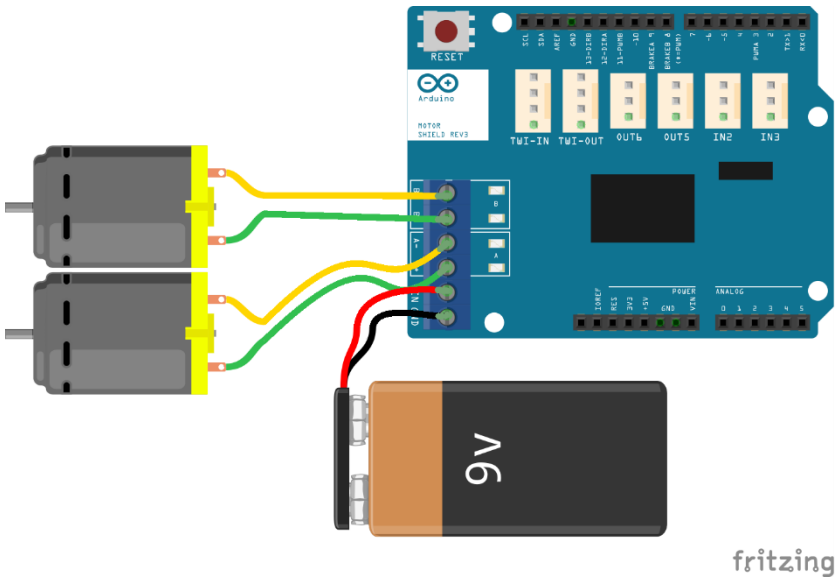
This shield has two separate channels, called A and B. Each one uses 4 of the Arduino pins to drive or sense the motor. In total, there are 8 pins used by this shield. You can use each channel separately to drive two DC motors or combine them to drive one bipolar stepper motor. The shield's pins, divided by channel are shown in the table below:

Function	pins per Ch. A	pins per Ch. B
Direction	D12	D13
PWM	D3	D11
Brake	D9	D8
Current Sensing	A0	A1

If you don't need the Brake and the Current Sensing and you also need more pins for your application, you can disable this feature by cutting the respective jumpers on the back side of the shield.

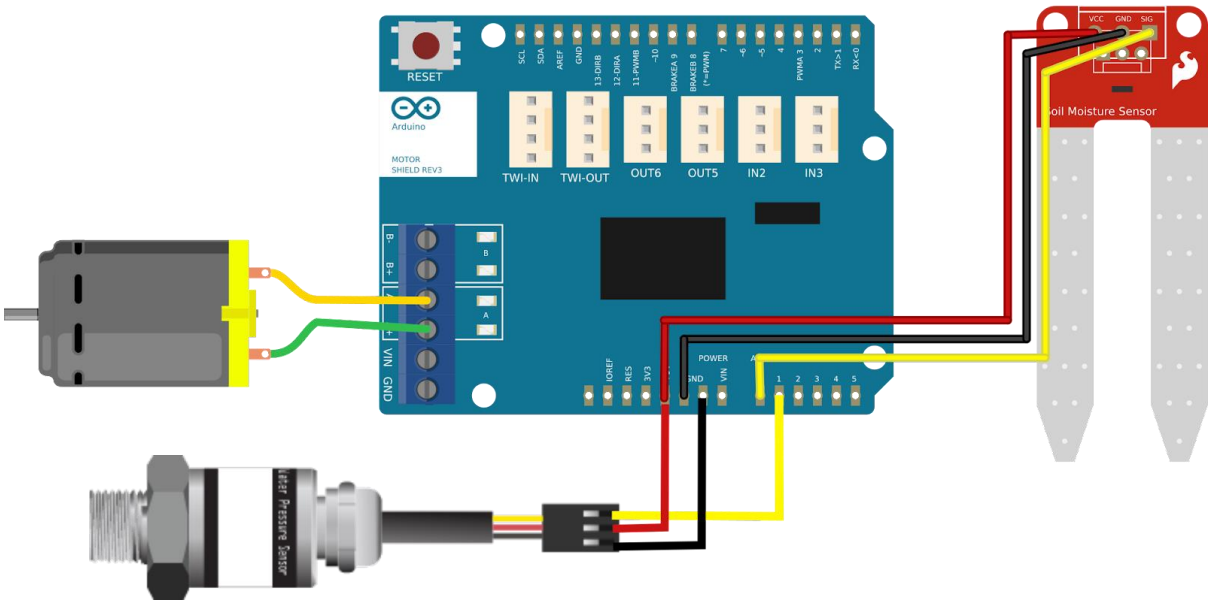
Co-funded by the Erasmus+ Programme of the European Union

This schema shows the motor connections. The motor shield can power the Arduino board, but this is not recommended.



This schema is for 2 motors, but we need only one (A), so pins D13, D11, D8, A1 can be used for other things.

This is the purpose schema. Arduino board is under the Arduino Motor Shield. Some pins can be changed if we need it for other purposes.



Notes:
Digital outputs can be used for opening valves if necessary.

Rather than powering the sensor through the 3.3V or 5V pins, can use a digital pin to power the sensor. This will prevent corrosion of the sensor as it sits in the soil.

Co-funded by the Erasmus+ Programme of the European Union

This is an example of code for reading moisture sensor. It can be used to read pressure sensor too changing the Analog pin:

```
int val = 0; //value for storing moisture value
int soilPin = A0;//Declare a variable for the soil moisture sensor

void setup()
{
 Serial.begin(9600);   // open serial over USB
}

void loop()
{
Serial.print("Soil Moisture = ");
//get soil moisture value from the function below and print it
Serial.println(readSoil());

//This 1 second timeframe is used so you can test the sensor and see it change in real-time.
//For in-plant applications, you will want to take readings much less frequently.
delay(1000);//take a reading every second
}
//This is a function used to get the soil moisture content
int readSoil()
{
  delay(10);//wait 10 milliseconds
  val = analogRead(soilPin);//Read the SIG value form sensor
  return val;//send current moisture value
}
```

Calibrating is difficult because it's not only the moisture, but the type of soil because the sensor is based on conductivity and the materials of the soil can change it. We recommend reading the serial output and trying with less and more water in the soil. Once you detect the minimum and maximum, you can use the map() function to get a normalized number. (https://www.arduino.cc/reference/en/language/functions/math/map/)

If we want to power on motor when moisture reach a value below a threshold it is easy to write in pins D12 and D3 a PWM value

PWM:
A digital output can change the state very quickly, so quickly that we can't see the change. But if we connect a LED to this pin, we can see less light. This is called "Pulse With Modulation" Arduino simplifies the code providing some pins with PWM and we only have to call the function analogWrite()

```
int val = 0; //value for storing moisture value
int soilPin = A0;//Declare a variable for the soil moisture sensor

void setup() {
  Serial.begin(9600);   // open serial over USB
 //Setup Channel A
 pinMode(12, OUTPUT); //Initiates Motor Channel A pin
 pinMode(9, OUTPUT); //Initiates Brake Channel A pin
}
```

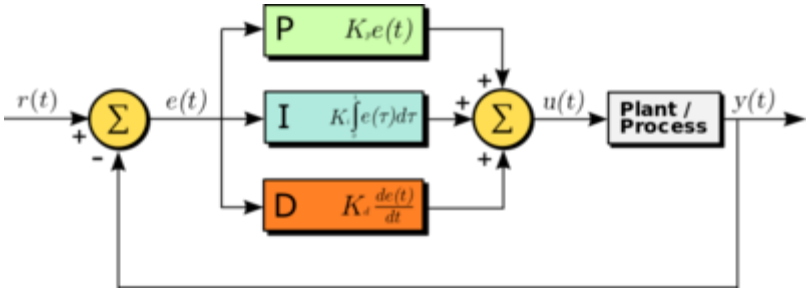
Co-funded by the Erasmus+ Programme of the European Union

```
void loop(){

Serial.print("Soil Moisture = ");
int moisture = readSoil();
Serial.println(moisture);

if(moisture < 100) {
 //Motor A forward @ full speed
 digitalWrite(12, HIGH); //Establishes forward direction of Channel A
 digitalWrite(9, LOW);   //Disengage the Brake for Channel A
 analogWrite(3, 255);   //Spins the motor on Channel A at full speed
}
delay(1000);//take a reading every second
}

int readSoil()
{
  delay(10);//wait 10 milliseconds
  val = analogRead(soilPin);//Read the SIG value form sensor
  return val;//send current moisture value
}
```

The above example shows how to fully run the pump. But the pressure can be excessive and we can control it with a pressure sensor. We can add this code to read pressure:

```
int press = 0;
int pressurePin = A1;

…

int readPressure()
{
  delay(10);//wait 10 milliseconds
  press = analogRead(pressurePin);//Read the SIG value form sensor
  return press;
}
```

We have to write a lower value to the 3 pin if the pressure is high. The problem is the reaction time of the pressure in the pipe. If the pressure is too low it can send a high value to the PWM pin and the pressure can be increased too quickly. The auto-calibration during the irrigation is difficult for a machine. Fortunately, the solution is apply a PID algorithm that is very well documented.



This is the schema of PID. It has three components:

- Proportional control: It controls the present correction of the speed.
- Integrative: It accumulates the past errors to correct the speed.

Co-funded by the Erasmus+ Programme of the European Union

- Derivative: It predicts the “future” errors to correct the speed.

It depends on the calibration of these three factors the smooth response to the pressure sensor.

This is an example of how to use the code for PID:

```
error = (pressure - IDEAL_PRESSURE);     // Proportional control
error = error / 10;
if (((long)error*(long)integral)<0)
   { integral=0; }  // integral Windup.
integral = integral + error; // Integrative
derivative = error - before_error; // Derivative
correction = Kp * error + Ki * integral + Kd * derivative;
pumpSpeed = speed_base  + correction;
before_error = error;  // for derivative
```

Kp, Ki, Kd are the PID calibration variables that have to be initialized after experimenting. We recommend to put 1 in all and start the calibration with Kp. When it works like the red line, we can touch the others to reach the green line.

Co-funded by the
Erasmus+ Programme
of the European Union

Heating system with temperature and time controls

Fig. 1 Final prototype

The project objective is to recover an old chemistry laboratory electric heater and provide it with a temperature and time controls. The control can be also used in industrial devices.
An ON-OFF control has been chosen because of its hardware simplicity and low cost and easily to get components.
All settings are performed by two rotary encoders with button effect clicking the knobs. Setting are displayed by a 2 row and 16 characters LCD.
As some parts of this circuit involve mains voltage, it's absolutely important the supervision of adequately trained instructors.

Hardware

The temperature measurement is performed by a DS18B20. It is a Digital Thermometer that provides 9 to 12-bit (configurable) temperature readings which indicate the temperature of the device. Information is sent to/from the DS18B20 over a 1-Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor to a DS18B20. Power for reading, writing, and performing temperature conversions can be derived from the data line itself with no need for an external power source. It has the following features:

- Unique 1-Wire interface requires only one port pin for communication
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components

Co-funded by the Erasmus+ Programme of the European Union

- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Zero standby power required
- Measures temperatures from -55°C to +125°C. Fahrenheit equivalent is -67°F to+257°F
- ±0.5°C accuracy from -10°C to +85°C
- Thermometer resolution is programmable from 9 to 12 bits
- Converts 12-bit temperature to digital word in750 ms (max.)
- User-definable, nonvolatile temperature alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system.

Instead of older and less flexible potentiometers, more modern rotary encoders have been chosen. Il the image below (fig. 2) it's possible to see both top and side view of the encoder module to use. In the top view are clearly visible all of the connection pins.
Fig. 3 represents both A (CLK) and B (DT) signals. Each rotation step causes A and B changing:

- a clockwise rotation makes A to change before B
- a counterclockwise rotation makes A to change after B.

Essentially, determining which switch changed states first is how the direction of rotation is detected. By swapping A and B pins, the rotation directions of increase and decrease will consequently be exchanged.

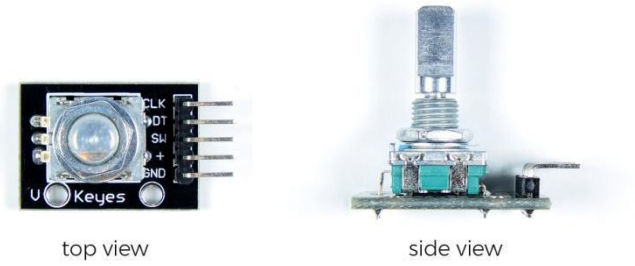


Figura 2 - Encoder view

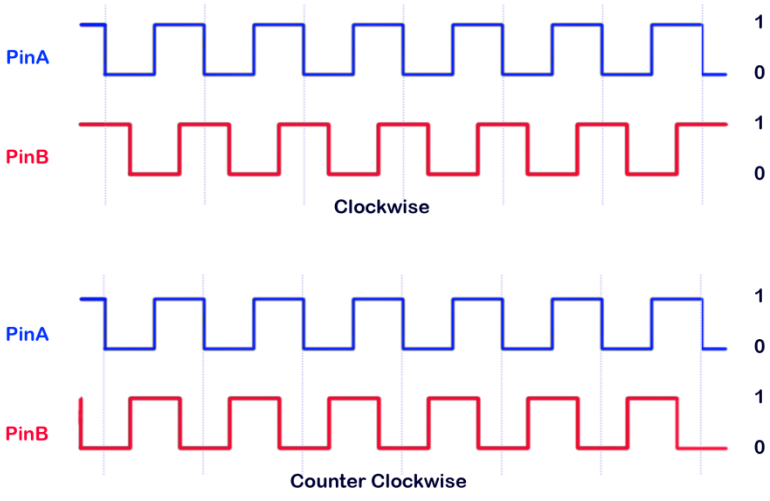


Fig. 3 - Rotary encoder signals

Co-funded by the Erasmus+ Programme of the European Union

This signal can be managed directly (using interrupt function to read the CLK pin). However, in our case, this task is performed by a library to simplify the program. Depending on the A and B pin connections, some corrections can be required. In the program (section *Encoders loop - knobs reading*) the read value is inverted (otherwise it's possible swap A and B pins connections) and divided by two (the library function read both rising and falling edges).

System elements summary

- Arduino Uno or compatible
- DS18B20 digital temperature sensor
- Temperature encoder (with click switch) from 20°C to 125°C
- Timer encoder (with click switch) from 00h00m to 99h00
- LCD 1602 I2C display
- Active buzzer
- Opto-isolated relay module 5Vdc – 250Vac, 10A
- DC power supply; input 240Vac, 40ma; output 12Vdc, 250mA;
- Fuse 250V, 10A retarded
- 230Vac controlled outlet (to connect the heater element)

Features of the system

- Temperature is ON-OFF mode controlled, using a opto-isolated relay as power interface.
- A global reset function (both time and temperature) is available clicking on the temperature knob.
- set time from 1 minute to 99 hours;
- set temperature from 20 to 125°C;
- temperature variation range aoround set temperature: +/- 0,5°C around it;
- an acoustic beep confirms that the temperature is reached;
- set and current temperature and time are displayed on LCD;
- time and temperature reset;
- temperature probe;
- maximum load of 2000W;
- protection fuse (10A);

How it works (refer to wiring diagram)

The system is able to keep almost constant (set temperature ± 0.5°C) the temperature of a substance. It is connected directly to a 230Vac socket and provide a on-off controlled socket to connect the plug of the heating device (maximum current=10A). To sense the temperature of the substance is used a DS18B20 digital sensor. It has a stainless-steel casing and can be immersed in liquid substances. To be more reactive, both time and temperature encoders are read by interrupt

Co-funded by the Erasmus+ Programme of the European Union

routines. It can be set both temperature (20÷125°C) and time through two switch-encoders as follow:

- Temperature can be set acting on the Temperature left knob (**2**), and seen on the display (**a**)
- Time can be set acting on the Time right knob (**3**), and seen on the display (**c**)
- Clicking on the Time knob, is possible to start heating; activation is confirmed by a “beep” sound (**5**)
- Also, display shows the current temperature (**b**), measured by the sensor (**1**)
- Remaining time is visible on the display as well (**d**)
- As the set temperature is reached, a double “beep” sound confirms it (**5**)
- When the timer is active a “T” symbol is showed on the display (**e**)
- When the heating element is active a “H” symbol is showed on the display (**f**), and the electric outlet on the device provide 230V to empower the heating element.

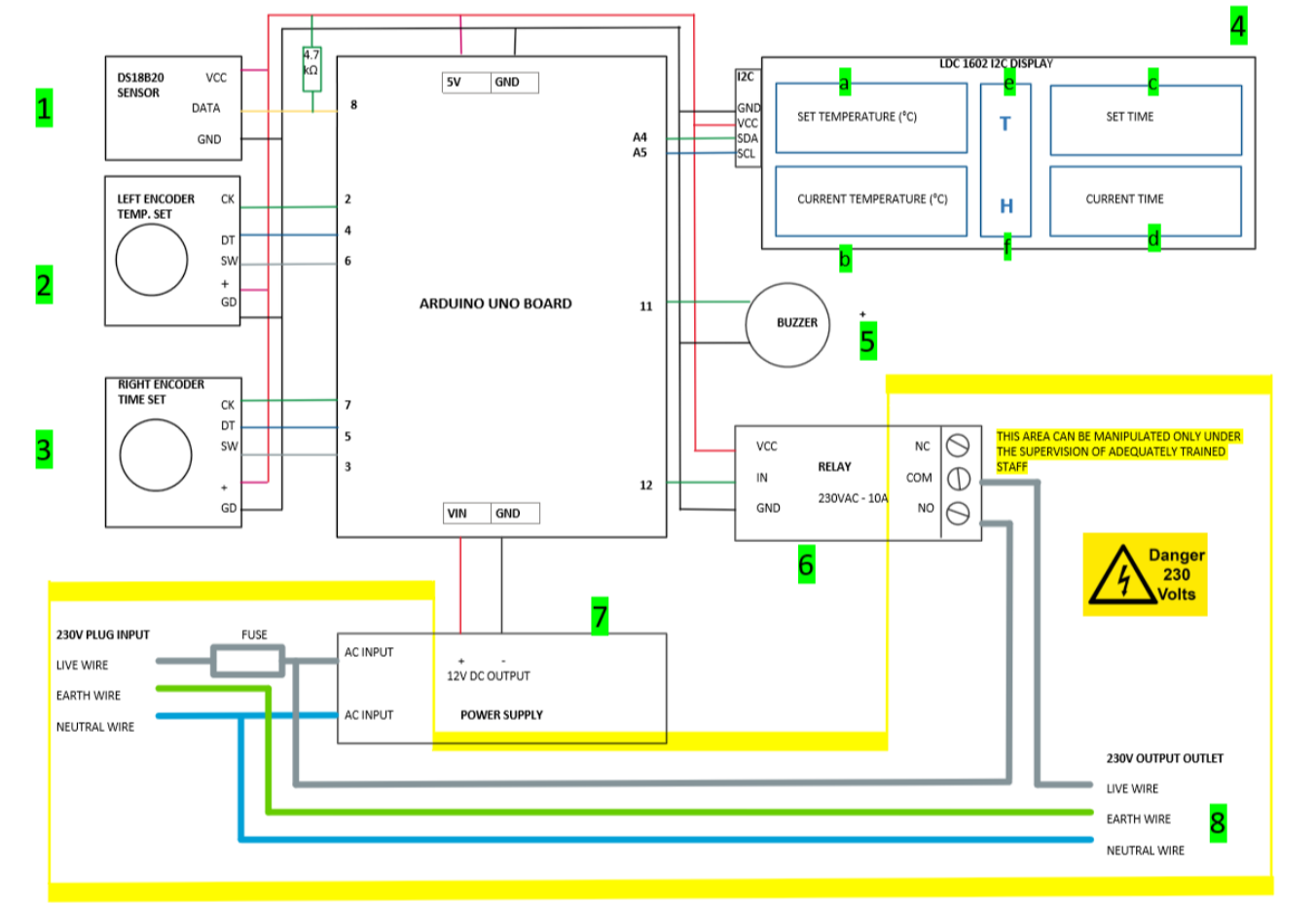


Fig. 4 - Wiring diagram

Co-funded by the Erasmus+ Programme of the European Union

Components detailing

1	Arduino Uno		Note Also different Arduino boards can be used.
2	DS18B20 temperature probe		A 4.7kOhm pull-up resistor between the data pin (yellow, connected to the digital input 8 of the Arduino board) and Vcc is required as the board HIGH level is 5V whereas the sensor output is 3.3V. To easily manage this digital sensor a proper library is used.
3	1602 display with I2C module		I2C interface is necessary to simplify the circuit and to reduce the number of Arduino pins required to manage the display. Libraries are used for both LCD display and I2C module.
3	Power adapter 230Vac-12Vdc - 250mA (AZ Delivery or equivalent)		Also other power adapters can be used. Arduino Uno accepts Vin from 7V to 12V.
4	Rotary encoder		A library is used to manage both the rotary encoders. Depending on the rotary module used it can be necessary to make some corrections in the code *(Encoders loop - knobs reading* section)
5	Opto-isolated relay module 5Vdc – 250Vac, 10A		Verify if your module input is active HIGH or LOW level. In the program below it is considered active HIGH. Vice-versa you'll need to modify properly the program in the "*HEATING CONTROL*" section.
6	Active buzzer - 5V		The classic 5V active buzzer often included in Auduino kits

Co-funded by the Erasmus+ Programme of the European Union

7	Fuse 250V - 10A Retarded		Circuit protection from overload. A fuse holder il also required.

Program structure

1. Libraries installing
   - Wire.h>
   - <LiquidCrystal_I2C.h>
   - <OneWire.h>
   - <DallasTemperature.h>
   - <Encoder.h>

2. Objects definition
3. Variables and constants definitions

Setup

- I/O configuration
- Serial Monitor configuration
- LCD initialization

Loop

1. Temperature settings and controls
2. Time settings and controls
3. Encoders loop - knobs reading
4. Knobs value reset
5. Timer control
6. Heating control
7. Alarm control
8. Serial monitor control (For debug only)
9. LCD control
   - LCD Set Temperature
   - LCD Set Time
   - LCD Remaining Time
   - LCD Flags

Buzzer functions

1. Beep
2. Beep2

Co-funded by the Erasmus+ Programme of the European Union

Program

```
//*************************** HEATER DS18B20 FULL PROGRAM ************************************

#include <Wire.h>
#include <LiquidCrystal_I2C.h>          // I2C LCD library
#include <OneWire.h>                    //Needed library for DS18B20
#include <DallasTemperature.h>                  //DS18B20 library
#define ONE_WIRE_BUS 8                  // DQ sensor pin
#include <Encoder.h>                            //Encoder library

Encoder knobLeft(2, 4);                         //Temperature control encoder
Encoder knobRight(3, 5);                        //Time control encoder
const int swLeftPin=6;                          //Reset pin
const int swRightPin=7;                         //Timer start pin
const int heatPin=12;                           //Heater relay pin
const int buzzerPin=11;                         //Buzzer pin

boolean swRightState=1;                         //Start timer button state
boolean swLeftState=1;                          //Temperature ad Time reset button state
float setTemp=0.0;                              //Set temperature
const float stepTemp=1.0;                       //Set temperature knob step
const int minTemp=20;                           //Minimum temperature admitted
const int maxTemp=125;                          //Maximum temperature admitted
const float dT=0.5;                             //Admitted range (+/-) around set temperature
float sensTemp;                                 //Sensor temperature value

boolean heatFlag=0;                     //Heating activation flag
boolean knobChangeFlag;                 //Changing set temperature flag
boolean timerFlag=0;                    //Activation timer flag

unsigned long maxMin=5940;              //Maximum minutes (99 hours)
unsigned long setMillis;                //Set milliseconds
unsigned long currentMillis;            //Current milliseconds
unsigned long startMillis;              //Milliseconds value when the timer starts
unsigned long elapsedMillis;            //Milliseconds elapsed from the timer start

unsigned int elapsedMinute=0;           // Minutes elapsed from the timer start
unsigned int remainingMinute;           //Remaining minutes
unsigned int currentMinute=0;           //Current minute displayed on the LCD
unsigned int currentHour=0;             // Current hour displayed on the LCD
unsigned int setHour=0;                 //Set hour
unsigned int setMinute=0;               //Set minutes

//Encoder pins and variables
long tempEncValue;                              //Temperature encoder value (left)
unsigned long timeEncValue;                     // Time encoder value in minutes (right)
long positionLeft  = 0;
long positionRight = 0;

OneWire oneWire(ONE_WIRE_BUS);                  // Set OneWire connection
DallasTemperature sensore(&oneWire);    // Sensor object declaration

LiquidCrystal_I2C lcd(0x27,16,2);               // set the LCD address to 0x27for the display

void setup()
{
  pinMode(swRightPin, INPUT_PULLUP);            //Input-Output configuration
  pinMode(swLeftPin, INPUT_PULLUP);
  pinMode(heatPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);

  Serial.begin(9600);                           // initialize the Serial Monitor
  lcd.init();                           // initialize the lcd
  lcd.backlight();
  sensore.begin();                              // initialize the sensor
  lcd.setCursor(10, 1);                         // set the cursor to column 10, line 1
  lcd.print("00h00m");                          // current time reset on LCD.
}
```

Co-funded by the Erasmus+ Programme of the European Union

```
void loop(){

 //Temperature settings and controls
 setTemp=minTemp+tempEncValue*stepTemp;                    //Temperature setting
 if(setTemp>maxTemp) knobLeft.write((maxTemp-minTemp)*2);  // Max set temperature control
 if(setTemp<minTemp) knobLeft.write(0);                    // Min set temperature control

 //Time settings and controls
 if(timeEncValue>maxMin) knobRight.write(0);               // Max set time control
 if(timeEncValue<0) knobRight.write(0);                    //Zero minutes control

 setHour=timeEncValue/60;
 setMinute=timeEncValue % 60;

 sensore.requestTemperatures();                            // Temperature sensor request
 sensTemp = sensore.getTempCByIndex(0);                    // Sensor temperature in °C

 //Encoders loop - knobs reading
 long newLeft, newRight;
 newLeft = knobLeft.read();                                //Temperature knob reading
 newRight = knobRight.read();                              //Time knob reading

 if (newLeft != positionLeft || newRight != positionRight) {
   tempEncValue=-newLeft/2;                                //Temp encoder correction *(-1/2)
   timeEncValue=-newRight/2;                               //Time encoder correction *(-1/2)
   positionLeft = newLeft;                                 //Temp. encoder position update
   positionRight = newRight;                               //Time encoder position update
   knobChangeFlag=HIGH;                                    //knobs changing flag activation
 }

 //Knobs value reset
 swLeftState=digitalRead(swLeftPin);                       //reset button reading
 if(swLeftState==LOW){                                     //if active => reset knobs value
   knobRight.write(0);
   knobLeft.write(0);
   timerFlag=LOW;                                          //Timer flag reset
   elapsedMillis=0;                                        //Elapsed time reset
   lcd.setCursor(10, 1);                                   // set the cursor to column 10, line 1
   lcd.print("00h00m");                                    // current time reset on LCD
   beep();

   }

 //TIMER CONTROL
 swRightState=digitalRead(swRightPin);                     //Start button reading
 if(swRightState==LOW){                                    //if active the timer is initialized
   startMillis=millis();                                   //start milliseconds
   setMillis=timeEncValue*60000;                           //set milliseconds
   timerFlag=HIGH;                                         //timer flag activation
   knobChangeFlag=HIGH;                                    //knob changing activation
   beep();
   }

 if(remainingMinute==0&&swRightState==HIGH) timerFlag=LOW;  //flag timer testing
 digitalWrite(LED_BUILTIN, timerFlag);                     //timer flag led on
 if(timerFlag==HIGH){
   elapsedMillis=millis()-startMillis;                     //Time check
   }

 elapsedMinute=round(elapsedMillis/60000);                 //Conversions for the LCD
 remainingMinute=timeEncValue-elapsedMinute;
 currentHour=remainingMinute / 60;
 currentMinute=remainingMinute % 60;


 //HEATING CONTROL
 if(sensTemp <= (setTemp - dT)&&timerFlag==HIGH)heatFlag=HIGH;   //Heater ON
 else if(sensTemp >=(setTemp + dT)||timerFlag==LOW)heatFlag=LOW; //Heater OFF
 digitalWrite(heatPin,heatFlag);                           //Relay pin is active HIGH
 //ALARM CONTROL
 if(knobChangeFlag==HIGH&&sensTemp>(setTemp-0.1)&&sensTemp<(setTemp+0.1)){
   beep2();
```

Co-funded by the Erasmus+ Programme of the European Union

```
  knobChangeFlag=LOW;
 }

//Serial monitor control (For debug only)
 Serial.print("Inc. temp. = ");
 Serial.print(tempEncValue);
 Serial.print(", Minuti = ");
 Serial.print(timeEncValue);
 Serial.print("\t");
 Serial.print(setHour);
 Serial.print(":");
 Serial.print(setMinute);
 Serial.print("\t");
 Serial.print("swR= ");
 Serial.print(digitalRead(swRightPin));

 Serial.print("\t Elapsed: ");
 Serial.print(elapsedMinute);
 Serial.print("\t Rem: ");
 Serial.print(remainingMinute);
 Serial.print("\t tF: ");
 Serial.print(timerFlag);
 Serial.print("\t kcF: ");
 Serial.print(knobChangeFlag);
 Serial.println();


// LCD CONTROL ----------------------------------------------------------------------------
 //LCD Set Temperature
 lcd.setCursor(0, 0);
 if (setTemp<100)lcd.print(0);
 if(setTemp>maxTemp)lcd.clear();
 lcd.print(setTemp,1);
 lcd.print("\xDF""C");
 //LCD Current temperature
 lcd.setCursor(0, 1);
 if (sensTemp<100)lcd.print(0);
 lcd.print(sensTemp,1);
 lcd.print("\xDF""C");

 // LCD Set Time
 lcd.setCursor(10, 0);
 if (setHour<10)lcd.print(0);
 lcd.print(setHour);
 lcd.print("h");
 if (setMinute<10)lcd.print(0);
 lcd.print(setMinute);
 lcd.print("m");

 //LCD Remaining Time
 if(timerFlag==HIGH){              //Remaining time is updated only with timer active
  lcd.setCursor(10, 1);
  if (currentHour<10)lcd.print(0);
  lcd.print(currentHour);
  lcd.print("h");
  if (currentMinute<10)lcd.print(0);
  lcd.print(currentMinute);
  lcd.print("m");
 }

 //LCD Flags
 lcd.setCursor(7,1);
 if(heatFlag==HIGH)lcd.print(" H ");                    //Heater flag "H"
 else lcd.print("   ");

 lcd.setCursor(7,0);
 if(timerFlag==HIGH)lcd.print(" T ");                   //Timer flag "T"
 else lcd.print("   ");
}

//======================== BUZZER FUNCTIONS ============================================
```

```
void beep(){                                    //single beep
  digitalWrite(buzzerPin,HIGH);
  delay(100);
  digitalWrite(buzzerPin,LOW);
  }
void beep2(){                                   //double beep
  digitalWrite(buzzerPin,HIGH);
  delay(100);
  digitalWrite(buzzerPin,LOW);
  delay(200);
  digitalWrite(buzzerPin,HIGH);
  delay(100);
  digitalWrite(buzzerPin,LOW);
  }
```

Testing

Fig. 5 - Final testing

Co-funded by the
Erasmus+ Programme
of the European Union

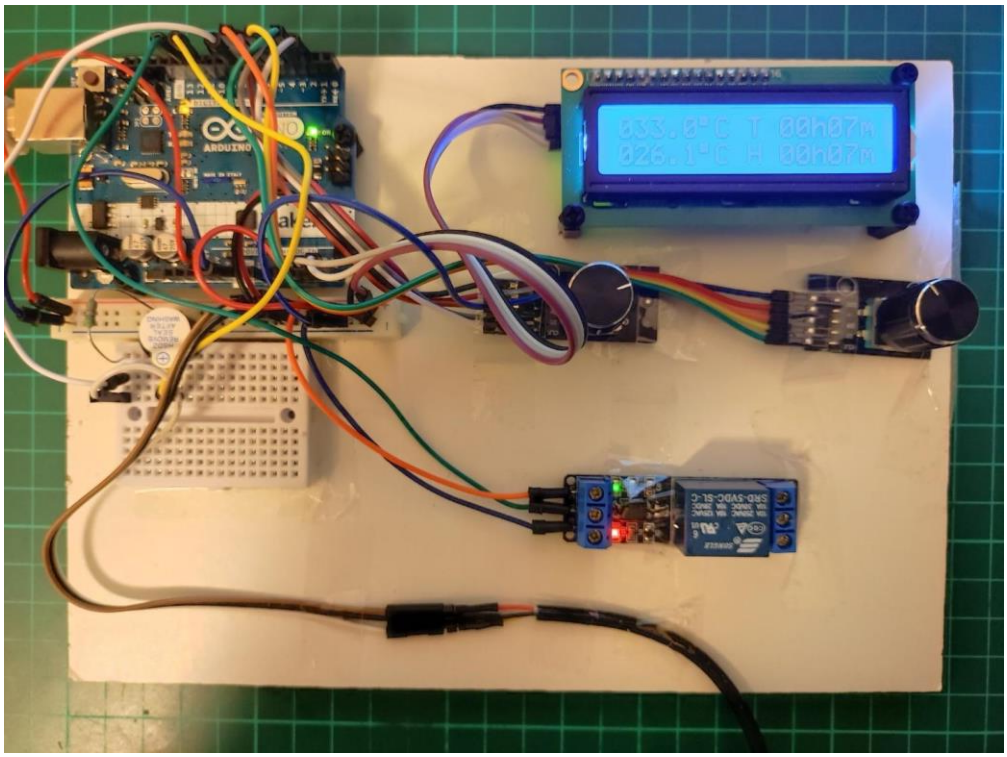


Fig. 6 - Early testing

During the early development phases, the circuit was tested section by section to verify if each of them worked properly.
Subsequently, the system was tested as a whole (except the mains voltage parts) to check for possible unwanted interactions.
In the end, the final prototype was assembled in a case and fully tested, including mains voltage section. To simulate the heating element, an incandescent lamp was used.
After all these tests, the device was used in a real-life scenario to do some experiments in a biology laboratory.

Co-funded by the
Erasmus+ Programme
of the European Union

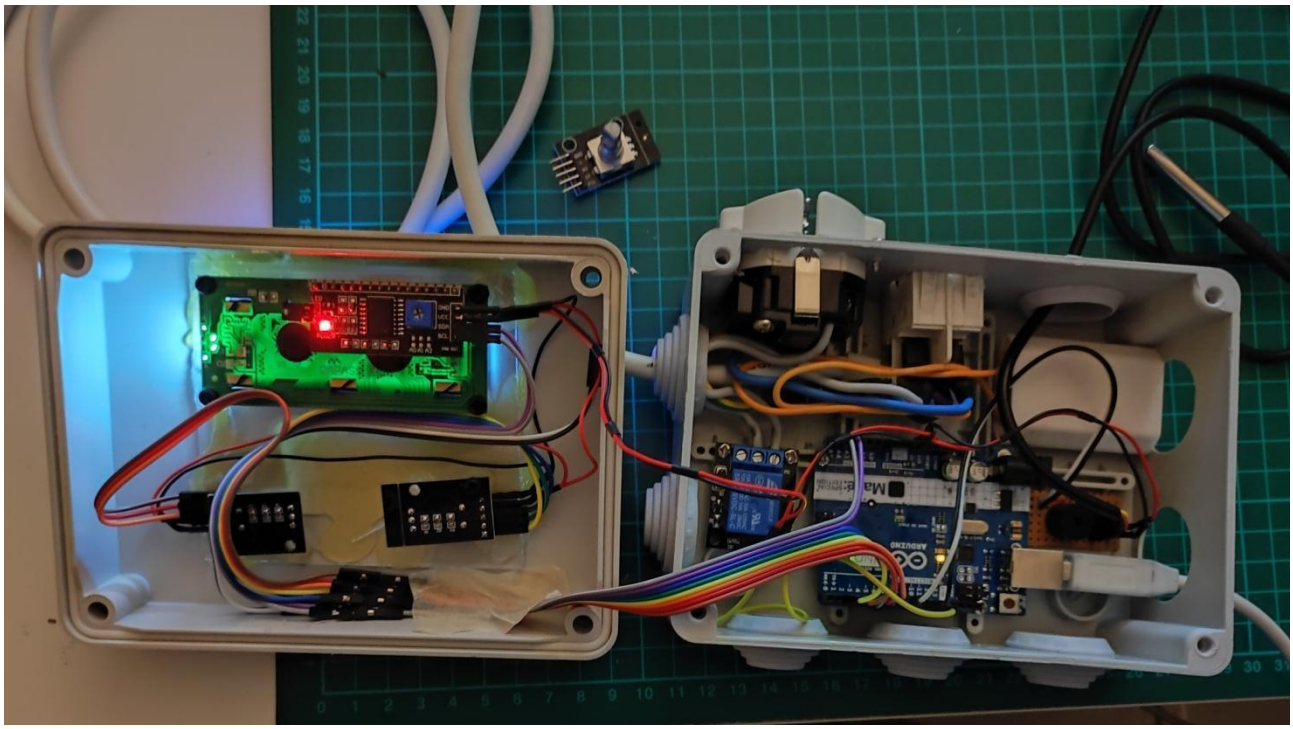


Fig. 7 - Almost complete final prototype

Problems and possible improvements

During the experiments in the laboratory with the real heater, the problem of a thermal inertia higher than expected emerged. This resulted in a temperature fluctuation of even a few degrees and much higher than expected.
This aspect is an interesting starting point for future improvements:
Replacement of the ON-OFF control with a more ready and precise PID control and consequently replacement of the traditional electromechanical relay with an alternating current solid state relay with zero-crossing detection.
It will obviously be necessary to adapt the part of the program relating to the heater control.

Bibliography

- https://www.arduino.cc/
- Dallas Semiconductors DS18B20 datasheet
- https://www.az-delivery.de/it/collections/kostenlose-e-books

Co-funded by the Erasmus+ Programme of the European Union

IV. Universal Measure Reader

From numeric display to digital information

In the context of recovering old machinery, it could be useful to read the measures provided by the old devices and make available these data to other software platform or systems, e.g.:

- storing data in cloud
- exporting data in Excel or CSV files
- automatically generating graphs or diagrams

This chapter describes the design and the implementation of a Universal Measure Reader (UMR), i.e., a component that can *read* measures from several different digital devices by analyzing images captured from the output displays and by apply optical character recognition techniques (Figure 1).

In order to build an UMR, two components are required:

- Hardware for capturing the images: it can be an Arduino or Arduino compatible board
- Software for analyzing the image and extracting, elaborating, storing, and presenting the data
- (Optional) Hardware for providing feedback:
- A green led switched on if the measure is in the expected range
- A buzzer for notifying errors or problems
- A switch for turning off or on a device in according with a rule based on the captured measure.

There are several possible choices both for hardware and software components, in this chapter will consider some of the most suitable.

Co-funded by the
Erasmus+ Programme
of the European Union

Structure of the chapter

This chapter is composed by the following sections:

- Hardware: it includes of a brief presentation of the hardware components used in the proposed projects.
- System Architecture: it includes some examples of system architectures for implementing a universal measure reader.
- Software Libraries: it includes a description of the software libraries that can be used for OCR.
- Projects
    - Project 1: Capture a measure every 30 seconds
    - Project 2: Capture a measure on a button press
    - Project 3: Switch on a green led on a *correct* measure
    - Project 4: Send measures to cloud
- Bibliography

Hardware

Among the several boards available on the market, that are able to grab a picture we have selected for our projects the ESP32-CAM board, because the following main reasons:

- It is fully programmable via Arduino IDE, so it can be exploited the same tools and knowledges used/acquired for the other projects described in this document.
- It is very cheap: typically, it costs less than 10 €. It is important in relation to renew old machinery to keep the costs low for keep the project profitable beyond environmentally friendly.
- It is quite easy to configure: In our project we have started from the base examples provided with the Arduino libraries and added further features and functionalities.

In the following sections are described the ESP32-CAM boards, a comparison of the models available on the market, the description of our preferred board and two tutorials about the Arduino IDE configuration and about the basic setup of the ESP32-CAM.

ESP32-CAM

The ESP32-CAM is a board that includes:

- ESP32-S chip: it is a dual-core microcontroller that combines Bluetooth and Wi-Fi wireless capabilities.
- OV2640 camera: it is an image sensor released in 2005 and although it is a 15-years-old technology is one of the most used sensors.
- MicroSD card slot: it supports microSD cards up to 4 GB.

Co-funded by the Erasmus+ Programme of the European Union

- 10 accessible GPIOs: only few GPIO can be really used because most are used for connecting the camera or the MicroSD card reader. For this reason, in some projects, we need to put beside this board an Arduino board.
- 4 MB PSRAM: it is used to buffering images from the camera.
- Reset Button: it is used for restarting the board.
- Flashlight LED: it is a very bright LED that can be used as flashlight.

Some of the ESP32-CAM boards does not include an USB-UART interface, so they cannot be directly connected to a PC for being programmed and they require a FTDI programmer (TTL to USB) or an ESP32-CAM MB programmer.

ESP32-CAM Comparison

There are several ESP32-CAM boards available on the marked, in (1) is reported a detailed article with a comparison among seven different boards. We report hereafter a summary of the comparison among four different boards.

Model	AI-THINKER	ESP-EYE	T-JOURNAL	TTGO T-CAMERA
				
PSRAM	4 MB	8 MB	X	8MB
SCREEN	X	X	0.91” OLED	0.96” OLED
MicroSD Card	✓	X	X	X
Microphone	X	✓	X	✓
Battery Connector	X	X	✓	✓
Built-in Programmer	X	✓	✓	✓
GPIO	10	X	4	X
Detailed Review	(2)	(3)	(4)	(5)
Cost Range	7 – 14 €	30 – 40 €	12 – 20 €	20 – 30 €

AI-THINKER

In the projects reported in this chapter we used the AI-THINKER board because it is a very good price-quality ratio:

- It is the cheapest board
- It has many more available GPIOs in comparison with the other boards
- It has a PSRAM module

The only drawback we found it is the missing of a USB-UART interface and so an external programmer is necessary for uploading the code.
We suggest combining the AI-Thinker board with the ESP32-CAM MB programmer (Figure 2): through this programmer it is easy both to upload code to the ESP32-CAM and provide power supply. The programmer includes also a useful reset button.
Typically, the ESP32-CAM MB is sold in combination with the AI-THIKER, otherwise it is available for less than 5 €.

Co-funded by the Erasmus+ Programme of the European Union

A detailed description of how uploading code to the ESP32-CAM with the ESP32-CAM MB programmer is provided in (6)

Figure 2 ESP32-CAM MB programmer

PINOUT

In Figure 3 is shown the PINOUT of the available pin of the AI-THINKER BOARD. There are other GPIOI that are directly connected with the camera and with the SD Card reader.

- If **IO0** is connected to **GND** the ESP32 is set in flashing mode and it is possible uploading code.
- **IO1** and **IO3** are used for serial transmission (TX and RX)
- The board support either 5V or 3.3V input voltage and can provide in output the same voltage

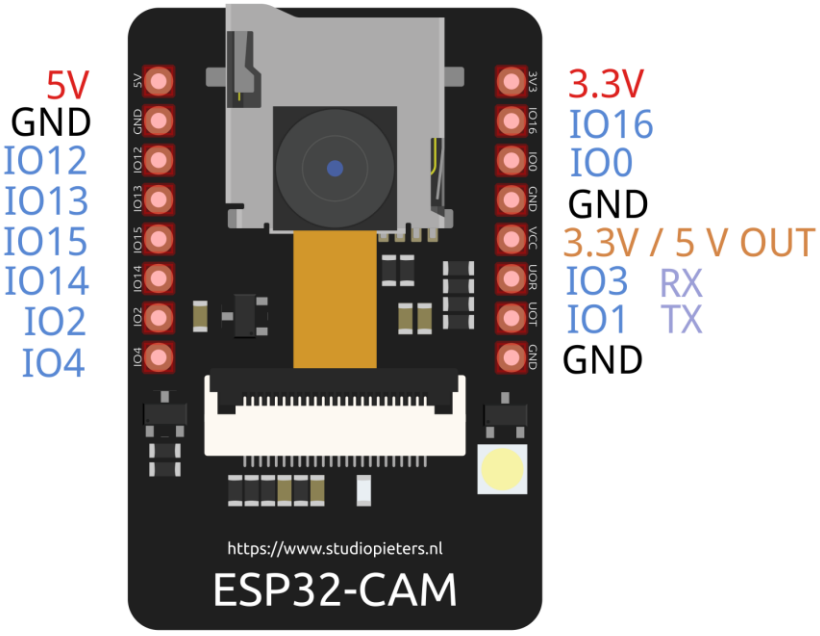


Figure 3 ESP32-CAM pinout

Co-funded by the Erasmus+ Programme of the European Union

Tutorial 1: Adding the ESP32 library to Arduino IDE

DIFFICULTY	Easy Medium Hard 
SOFTWARE REQUIREMENTS	Arduino IDE installed on Windows, Linux or macOS
HARDWARE REQUIREMENTS	None: in this phase it is not necessary to connect the ESP32-CAM board
TIME	about 5 minutes

ESP32-CAM board are fully compatible with the Arduino IDE[1] (7) and to include the EPS32 libraries and example it is necessary to perform the following steps:

1. Access to the Arduino IDE preferences through the menu: File Preferences

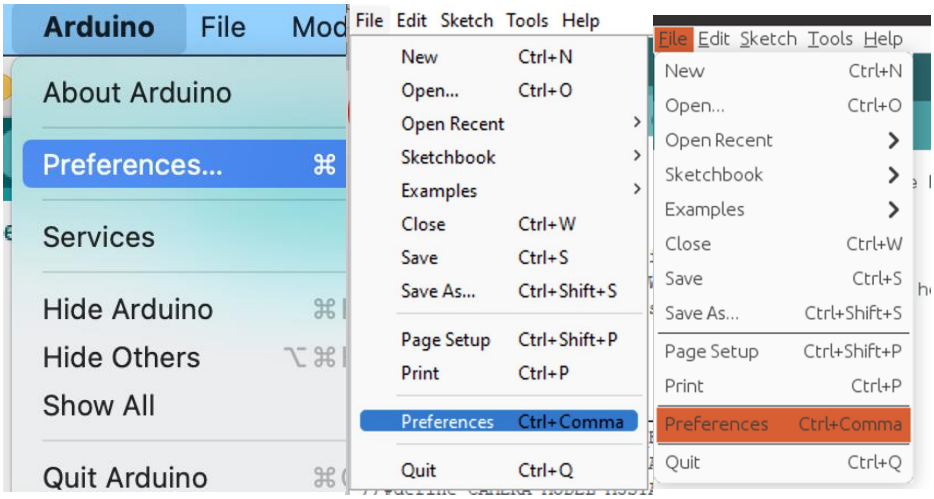


Figure 4 Arduino Preferences on A) macOS B) Windows C) Linux (Ubuntu 20.04)[2]

[1] In our screenshots we used the version 1.8.19 of Arduino IDE. In the time we are writing is almost available the next version of Arduino IDE (

[2] Since the Arduino IDE presents a user interface very similar among the 3 different operative systems, hereafter we include only one screenshot relative to Linux or Windows.

Co-funded by the Erasmus+ Programme of the European Union

2. Add the URL https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json to the Additional Boards Manager URLs section of the preferences dialog.

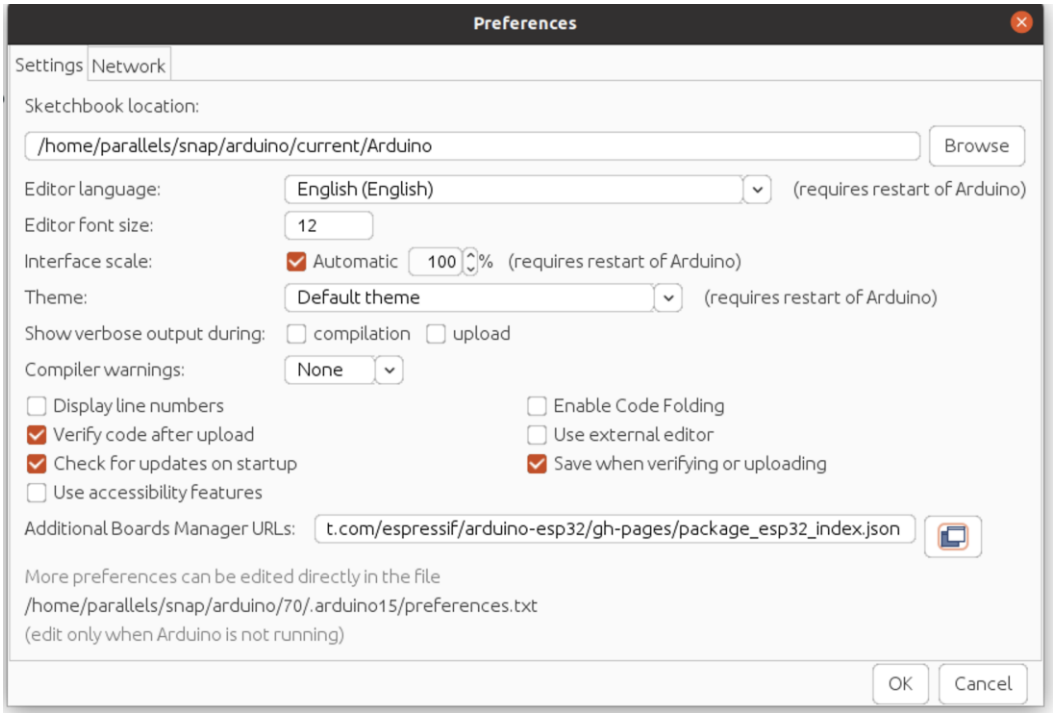


Figure 5 Additional Boards Manager URLs configuration

3. Open the Boards Manager through the Tools / Board / Boards Manager menu.

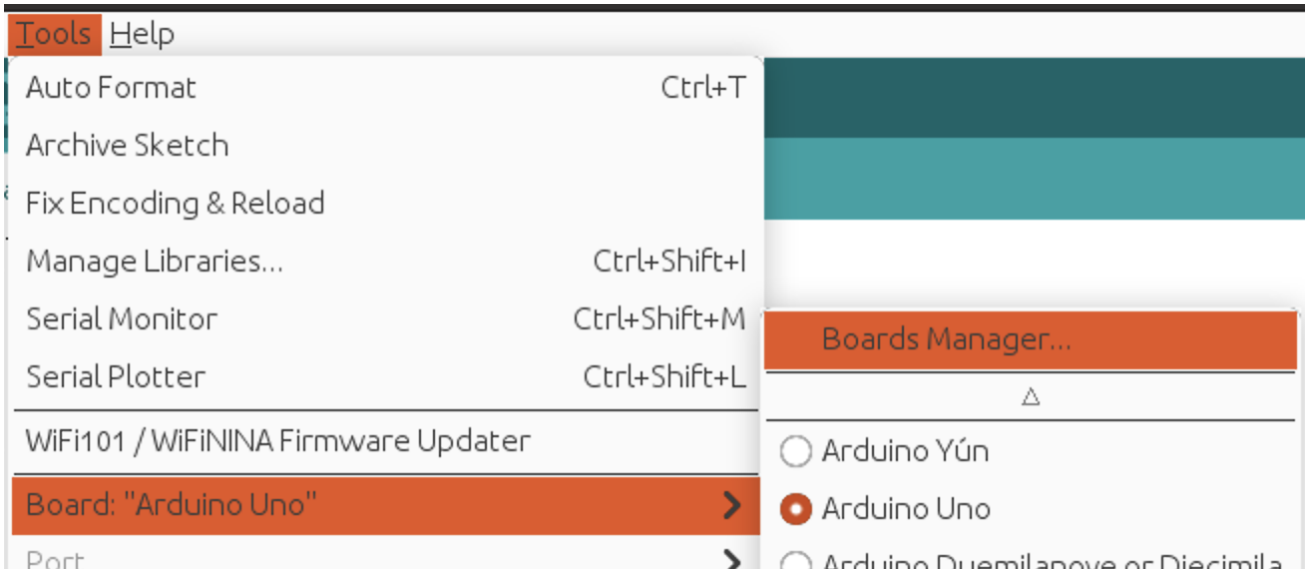


Figure 6 Boards Manager Menu

Co-funded by the Erasmus+ Programme of the European Union

4. Search and install the package esp32 by **Espressif** Systems.

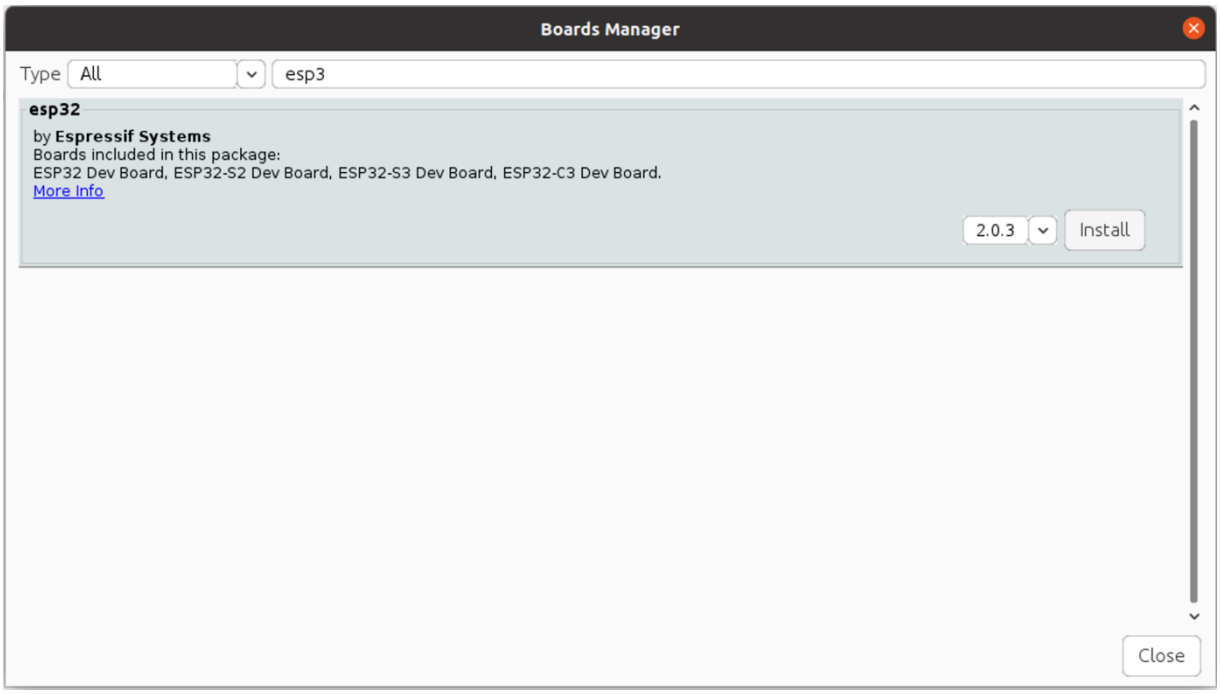
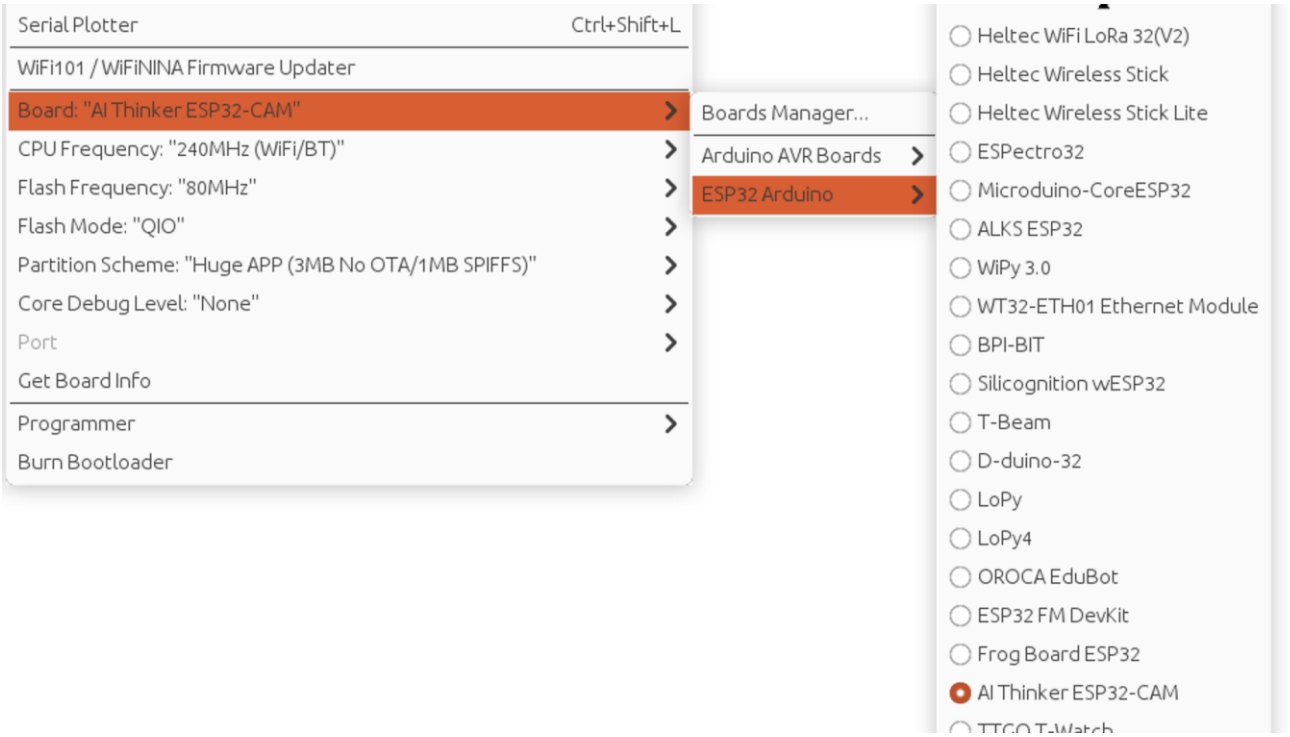


Figure 7 esp32 package

5. Select your board from the menu: Board ⍰ ESP32 Arduino. In our case we selected AI Thinker ESP32-CAM



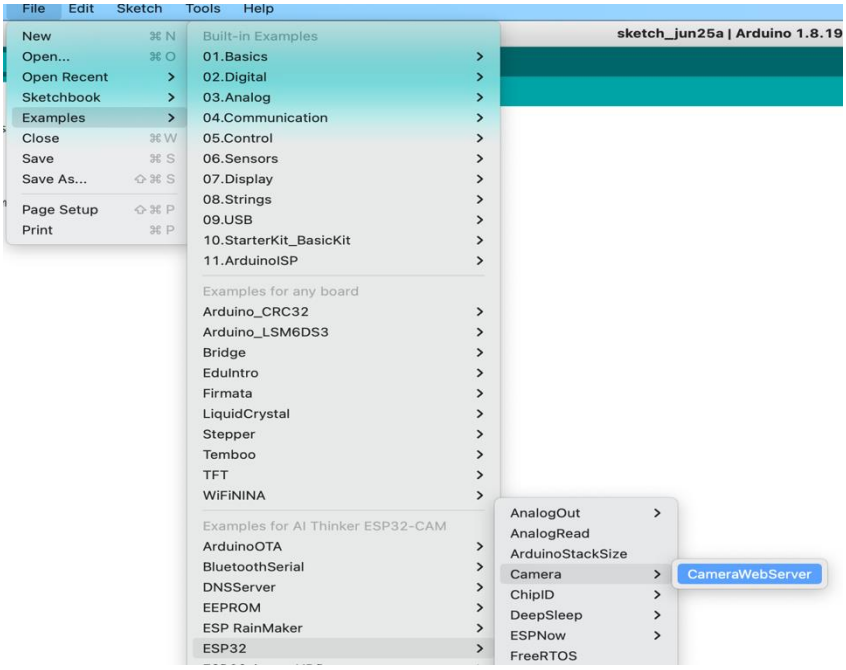
Co-funded by the
Erasmus+ Programme
of the European Union

Tutorial 2: Configuration of the ESP32-CAM board

DIFFICULTY	Easy Medium Hard 
SOFTWARE REQUIREMENTS	• Arduino IDE installed on Windows, Linux or macOS • ESP32 library installed on Arduino IDE (see Tutorial 1: Adding the ESP32 library to Arduino IDE)
HARDWARE REQUIREMENTS	• ESP32-CAM board • Programmer e.g., ESP32-CAM MB
TIME	about 10 minutes

The purpose of this tutorial is to show how the ESP32-CAM Board can be configured starting from the working example included in the Espressif package (see Tutorial 1: Adding the ESP32 library to Arduino IDE)

1. Once you have selected your ESP32-CAM Board through the Tools / Board / Boards Manager menu, you can open the CameraWebServer example through the File / Examples / ESP32 / Camera menu.



2. The CameraWebServer example create a simple web service that allows to interact with the camera. The sketch is composed by 4 files:

Co-funded by the Erasmus+ Programme of the European Union

- **CameraWebServer.ino**: it is the main Arduino file that contains the base configuration for selecting the correct board model and to initialize the Wi-Fi connection. This code starts a web server that host a simple web page that allows capturing picture or starting a video stream.
- **camera_pins.h**: it defines in according with the selected board model the pinout mapping. This file is very useful as GPIO reference for other Arduino Sketches using the ESP32-CAM.
- **camera_index.h**: it initializes the index_ov2640_html_gz variable with bytes of the gzip compressed page that will displayed by the web service. It is possible to upload a custom html page by compressing the html file with gzip (Dynamic Huffman Coding), splitting in bytes and convert to hex array. This process is described in detail in (8).
- **app_httpd.cpp**: it contains the implementation of the web service. If you need to add some handlers to the web service, you need to modify this file. The code is quite complex, so in order to understand all the functions it is required very good skill in c or c++ coding. Nevertheless, you can use most of the functions as they are, and only add slight changes. We will follow this approach in the projects described in the next sections.

3. To Set the ESP32-CAM board model, it is necessary to uncomment the define line in CameraWebServer.ino file. In our case, we uncommented the definition of CAEMRA_MODEL_AI_THINKER. This definition is used to correctly setting up the GPIOs mapping and for enabling.

```
// ===================
// Select camera model
// ===================
//#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
//#define CAMERA_MODEL_ESP_EYE // Has PSRAM
//#define CAMERA_MODEL_ESP32S3_EYE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
//#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
//#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
//#define CAMERA_MODEL_M5STACK_UNITCAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
//#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM
```

4. The ESP32-CAM board has native wireless connetivity: there are both WI-FI and Bluetooth capabilities. It is necessary to configure both the ssid and password for connecting the board the company/school network. You need to modify the following lines

```
// ============================
// Enter your WiFi credentials
// ============================
const char* ssid = "**********";
const char* password = "**********";
```

5. You need to verify and upload[3] the modified code to the ESP32-CAM board. You must select the correct serial port from the Tools Port menu. If the configuration is Ok, you can read on the serial monitor an output similar to the below figure.

```
rst:0x8 (TG1WDT_SYS_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1344
load:0x40078000,len:13516
load:0x40080400,len:3604
entry 0x400805f0

.
WiFi connected
Camera Ready! Use 'http://192.168.1.4' to connect
```

6. The started web server is available at the address displayed in the Serial Monitor just because the following lines of code:

```
Serial.println("");
Serial.println("WiFi connected");

startCameraServer();

Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP());
Serial.println("' to connect");
```

7. The web page depicted in will be displayed:

[3] To upload the sketch it is necessary to connect the ESP32-CAM board to a programmer and the programmer to the PC.

Co-funded by the Erasmus+ Programme of the European Union

Figure 8 ESP32-CAM configuration web page

The web page summarizes all the ESP32-CAM capability:

- Capture an Image
- Start a video Streaming
- Change the image resolution
- Set the image quality, brightness, and contrast
- Flip and Mirror the image
- Activate the face detection and recognition

In our experience, even if the embedded face detection and recognition features seem working, they are not reliable for access security purposes. In other words, we will not suggest exploiting these capabilities for granting access to systems or environments.

In Figure 9 is shown an example of streaming with the maximum resolution.

Co-funded by the Erasmus+ Programme of the European Union

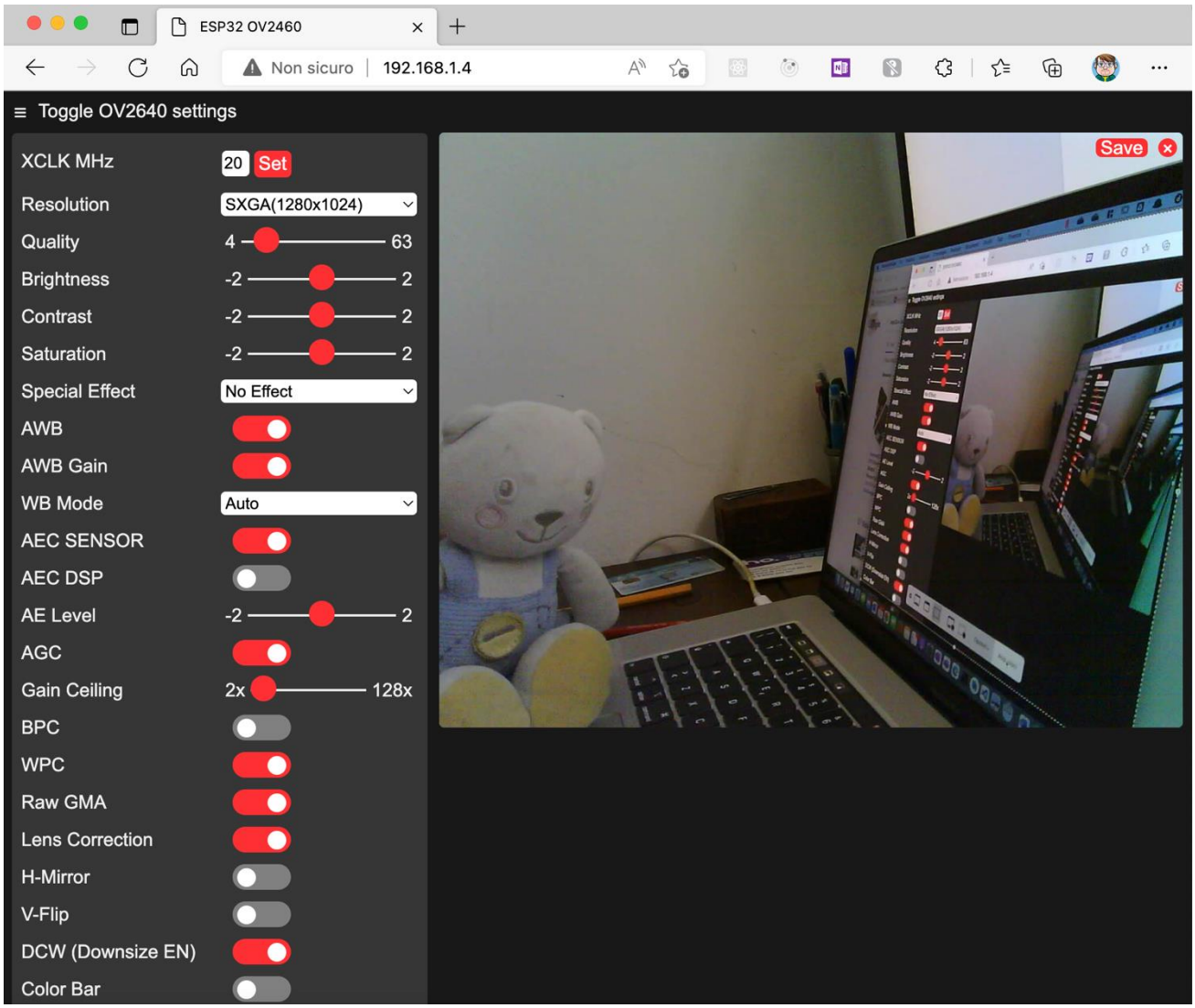


Figure 9 ESP32-CAM Streaming Video with maximum resolution

Co-funded by the
Erasmus+ Programme
of the European Union

System Architectures

The proposed Universal Measure Reader (UMR) is composed by both hardware and software components, and it is important to design which ones oversees implementing specific functionality.

The UMR provides the following main functionality:

- Grab a picture of the device display
- Analyze the picture and “understand” the measure via OCR techniques
- Store data and provide them to the user
    - Display the date in a web application
    - Export the data in Excel or CSV file
- Provide feedback via hardware notifications, e.g., a green led, a buzzer or mini-lcd display.

To achieve these objectives, there are several possible system architectures that can be chosen case by case in relation to the context and to the kind of old device that would be modernized.

Please note that the images included in the following sections represent a very high level of abstraction, while more detailed technical schemas will be provided in Projects section (page).

Co-funded by the
Erasmus+ Programme
of the European Union

Components

The UMR can be implemented using the components described in the below table:

COMPONENT	TYPE	DESCRIPTION
ESP32-CAM	Hardware	It is described in detail in previous ESP32-CAM section (page). Wireless connectivity Low Resolution Camera GPIO available for input (e.g., buttons) and output (e.g., leds) Face recognition (not used)
OCR server	Software	It is a custom web server that receives images and provide the "understood" measure. It exploits the software library described in the SOFTWARE section.
OCR cloud service	Software	It is an alternative to the above-mentioned OCR server. When the open-source tools exploited by the OCR custom server are not enough accurate for "understand" the measure, a cloud OCR service can be used.
Arduino boards	Hardware	If the ESP32-CAM GPIOs are not sufficient, it is possible to place beside also an Arduino board, that can: - drive complex output such as lcd display, led, buzzers - send information to the Arduino Cloud Service - manage the ESP32-CAM configuration via inputs: e.g., combination of buttons and potentiometers
Arduino Cloud Service	Software	It is a cloud service for storing Arduino IOT device data and display appealing dashboards.
Buttons, Leds, wires, Resistors, LCDs, potentiometers, etc.	Hardware	They are used in combination with ESP32-CAM or Arduino boards for providing inputs and outputs.

Co-funded by the Erasmus+ Programme of the European Union

Architecture 1: Software First System (SFS)

In this architecture the main component of Universal Measure Reader is the software module that perform most of the functionality. As depicted in Figure 10, the software module gets the picture of the measure from the ESP32-CAM boards, analyzes the image, and provide the results to user through a web page.
This design minimizes the hardware implementation: basically, the ESP32-CAM board it is only used for providing pictures like it was an external camera, because most of the work is done by the OCR server software.

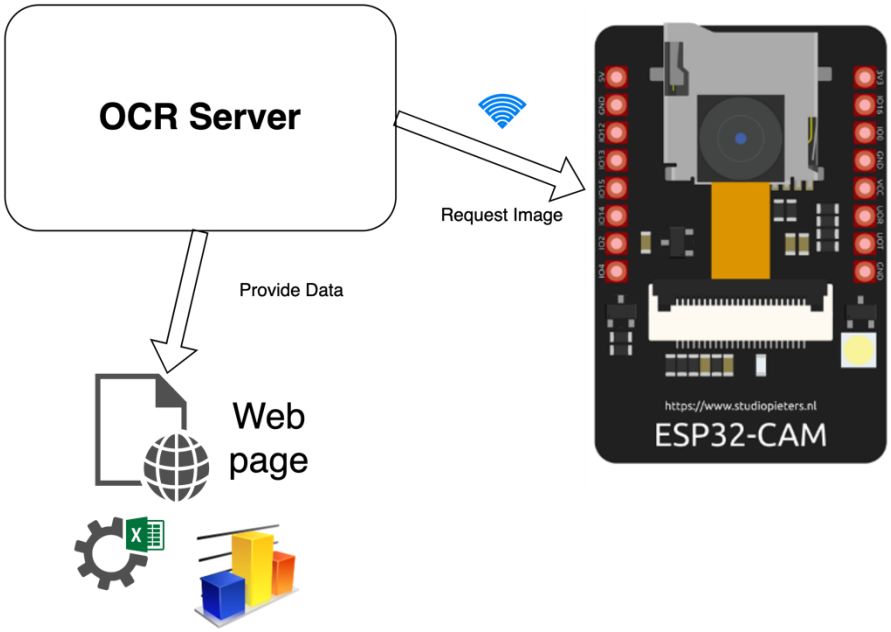


Figure 10 SFS Architecture

This kind of architecture is exploited by Project 1: Capture a measure every 30 seconds.

Co-funded by the Erasmus+ Programme of the European Union

Architecture 2: Hardware First System (HFS)

The Hardware First System Architecture privileges the hardware modules instead of the software ones. Figure 11 HFS Architecture shows an example of HTS architecture, where:

1) The user presses a button
2) The ESP32-CAM board reacts to the button press event and then it grabs a pictures and sends it to the OCR Server
3) The OCR Server analyses the image and provides the recognized measure to an Arduino board
4) The Arduino board receives the measure and notify it as an output (lcd display, led)

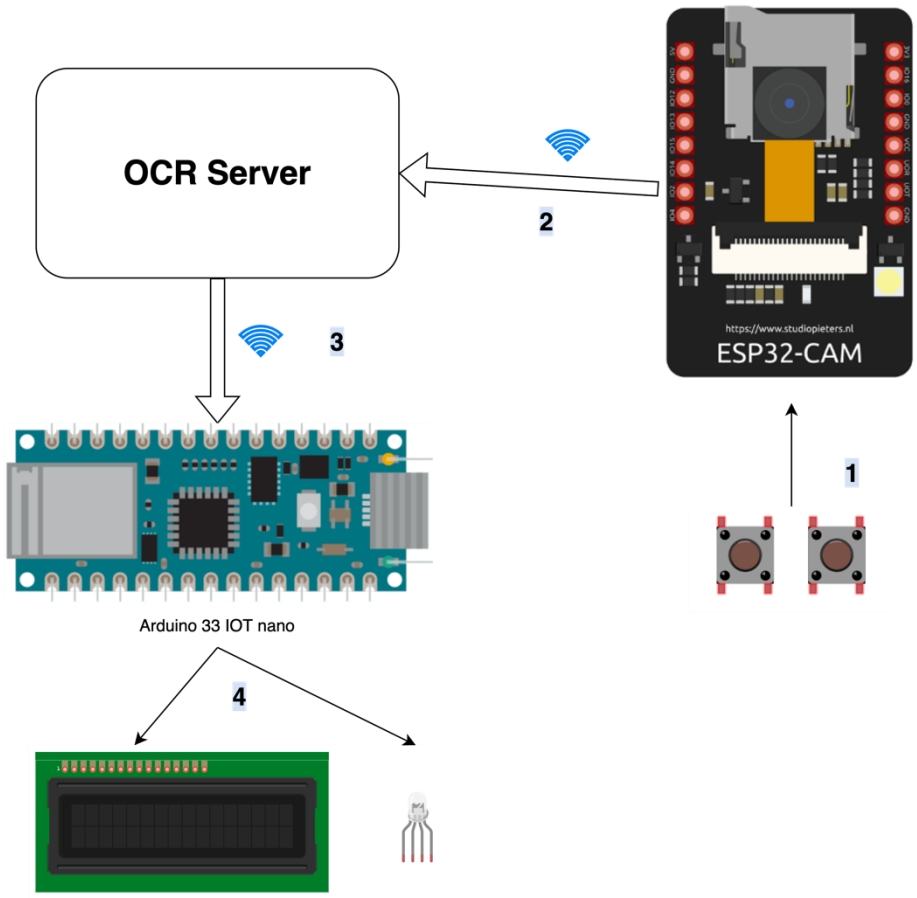


Figure 11 HFS Architecture

In this schema the OCR Server is lighter and perform two main functionalities:

- Image analysis
- Routing the output to another board

The second board is necessary because unfortunately, the EPS32-CAM board has few available GPIOs and so it is not possible to drive output directly from it.

Co-funded by the Erasmus+ Programme of the European Union

Architecture 3: Cloud based System (CLS)

Nowadays, the computational capacity is moved more and more out from the local hardware resources to huge external provider services, commonly named Cloud services.
As described in SOFTWARE section (page) there are sever Cloud providers and most of them offers AI OCR services.
There are at least two possible system architectures that involve cloud services in UMR:

- Cloud Only: the custom OCR service is completely substituted by cloud services
- Cloud with Middleware: the custom service delegates some functionalities to Cloud services but is the orchestrator of the system.

Cloud Only

As depicted in Figure 12, in this proposed architecture the ESP32-CAM directly exploits the Cloud Services.

It is feasible but not easy programming the ESP32-BOARD for establishing a connection with the selected Cloud service. It depends on provider to provider and from the APIs that they offer, but typically the board can exploit both Javascript API and REST services.
The Javascript API uses the computational capacity of the client (i.e., computer connected to the ESP32-BOARD service), while the REST service uses the (low) capacity of the board.

Also, the other board (in our example an Arduino 33 IOT) must be connected to Cloud service in order to be controlled.

An example of interaction is composed by the following steps:

1) The user presses the button
2) The ESP32-CAM board
    a. Capture an image
    b. Contact the Cloud service and send the Picture
3) The cloud service
    a. performs the OCR
    b. notifies the measure to the Arduino Board
4) The Arduino board
    a. Receives the measure
    b. Displays the output (e.g., using an LCD or a led)

Co-funded by the Erasmus+ Programme of the European Union

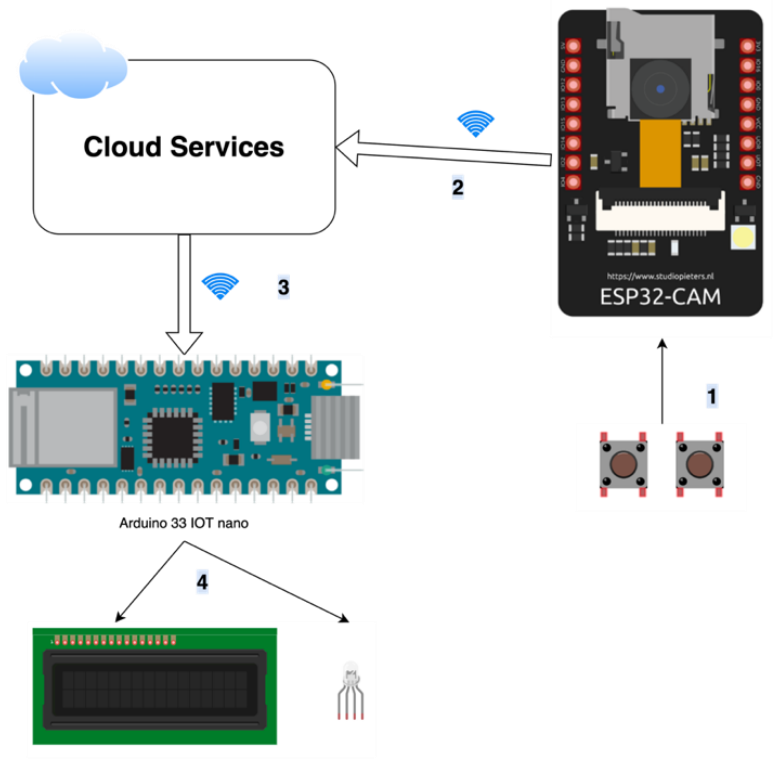


Figure 12 Cloud Only Architecture

Cloud with Middleware

Figure 13 shows the architecture involving cloud and a software middleware that mediates the communications between the ESP32-CAM board and cloud services and other boards.
In this case the middleware is a “smart” rooter of messages and notification, while the main computational capacity is delegate to cloud services.
An example of interaction is composed by the following steps:

1) The user presses the button
2) The ESP32-CAM board
    a. Capture an image
    b. Send the image the Middleware
3) The Middleware
    a. Exploits the cloud service for performing the OCR
    b. notifies the measure to the Arduino Board
4) The Arduino board
    a. Receives the measure
    b. Displays the output (e.g., using an LCD or a led)

Co-funded by the
Erasmus+ Programme
of the European Union

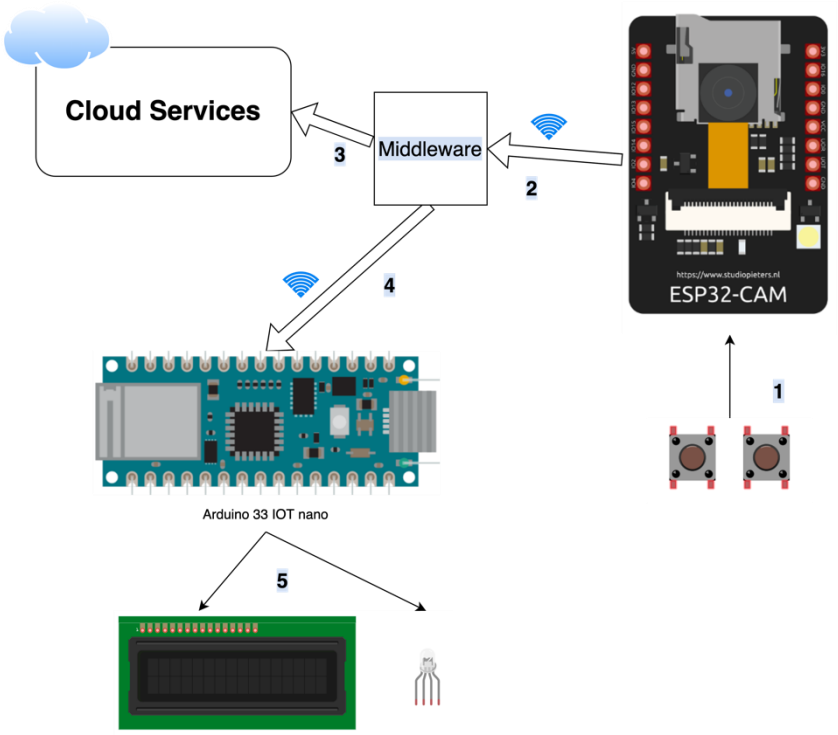


Figure 13 Cloud with Middleware

Co-funded by the
Erasmus+ Programme
of the European Union

Software

This section describes some useful software library that can be exploited for performing the measure recognition and for implementing the communication infrastructure (web services) between the components of the Universal Measure Reader.

Although it is possible to implement a Universal Measure Reader in most of the programming language, we chose in this guide a unique language: Python.

Python has been selected because it has a smooth learning curve and can be exploited also by not expert programmers. On top of that, the base of Python coding is taught in both VAT schools and high schools.

Optical Character Recognition (OCR)

The OCR systems are software applications that are able to recognize text characters in images or movies. They belong to the following fields of the AI:

- Image recognition
- Pattern recognition
- Artificial Vision

The origin of OCR systems is not recent (9): in 1927 the physicist Emanuel Goldber invented a machine that could convert printed text into telegraph code.

In the 1950s the first commercial OCR systems were developed; they were mainly used to digitalize coupons and postal addresses.

In late 1960s the protagonist of the OCR system scene was the Optical to Tactile CONverter, mostly known as Optacon (10). It is an electro-mechanical device that allows blind people reading printed material. The device is composed by two components: a scanner for reading the text and pad that “translate” the words in vibrations felt on the fingers tips.

From 1970s to 2000s the OCR technologies expanded more and more: many companies were found, and their application fields were increased, e.g.:

- Assistive technologies for blind and visual impaired people
- Address and Postal code detection for automatic mail routing
- Old newspapers, books, and magazines digitalization
- Form recognition
- Security: license plate number detection

Since 2010s, many software applications natively include OCR capabilities and most of the cloud vendors provide API services or SDK offering advanced OCR functionalities.

Co-funded by the Erasmus+ Programme of the European Union

In the next sections it is described how to exploit both Tesseract, one of the main open source OCR application, and the Microsoft Azure Computer Vision cognitive services, a very good and not so expense OCR cloud service.

Tesseract

Tesseract is without any doubts the most accurate **open-source** OCR engine. It was developed in the 1980s by Hewlett-Packard, the in 2005 it was released as released as open source. Since 2006 the development of Tesseract has been supported by Google, that integrates it in its cloud services (e.g., google drives, gmail, google cloud).

Installing Tesseract

In our project we make us of Tesseract python libraries, therefore it is necessary to install Tesseract engine in the server (or PC) where the application will be deployed.

The detailed tesseract installation documentation is available in (11); we report hereafter a brief summary.

On Windows

The windows installer is not directly provided in the official Tesseract-OCR github repository, but it is hosted by Mannheim University Library (UB Mannheim) at the following url: Home · UB-Mannheim/tesseract Wiki · GitHub

1. First of all, it is necessary to select the installer for the correct version of Windows (32 or 64 bit)[4]

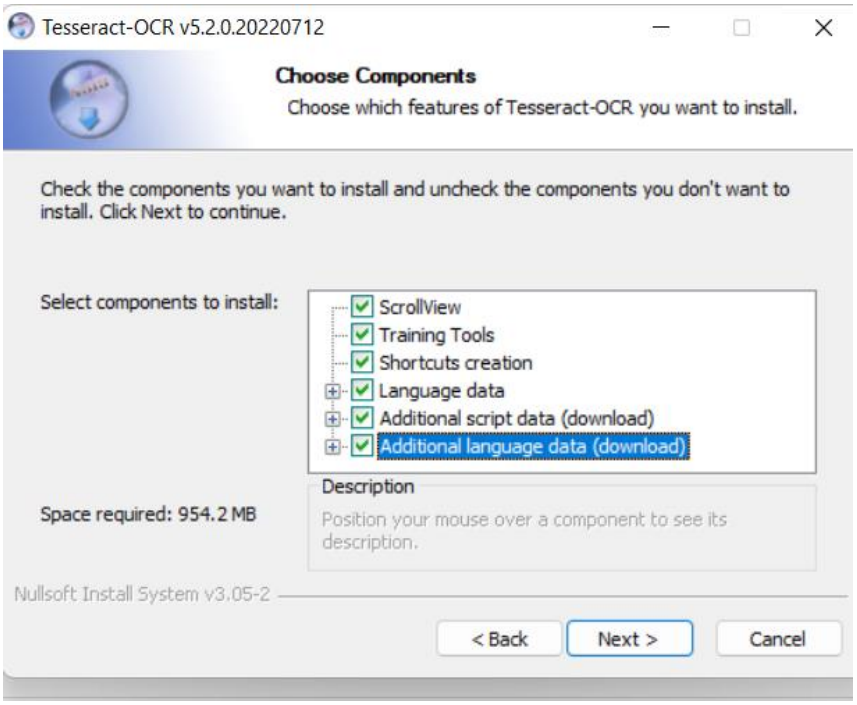
   The latest installers can be downloaded here:

   - tesseract-ocr-w32-setup-v5.2.0.20220712.exe (32 bit) and
   - tesseract-ocr-w64-setup-v5.2.0.20220712.exe (64 bit) resp.

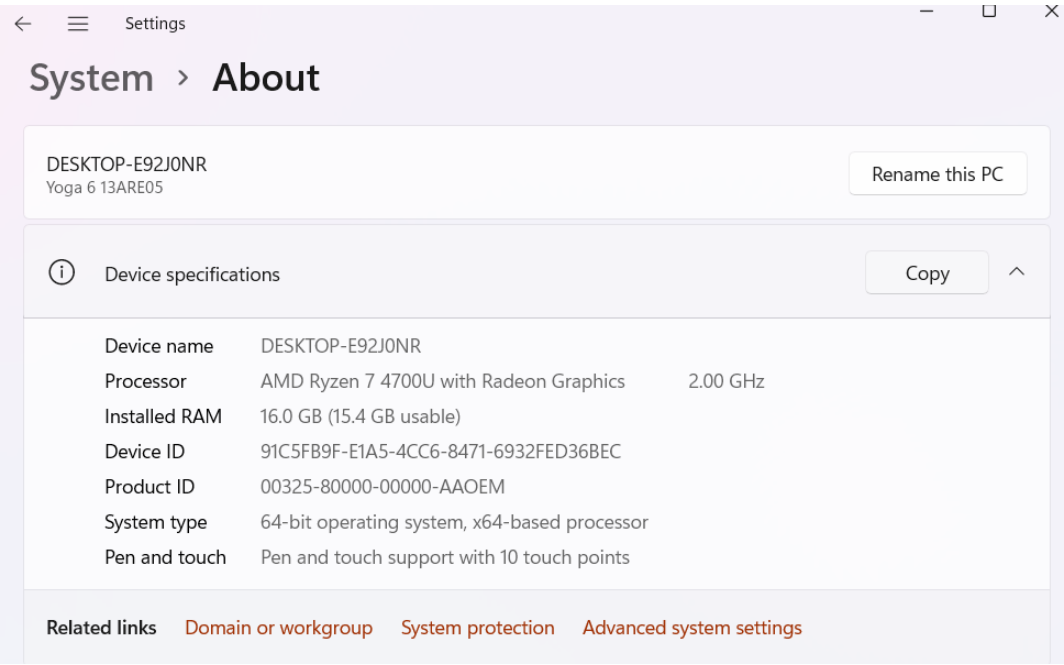
2. We suggest selecting all the optional features during the installation.

[4] To understand the system type, you need:
- Open the Start menu, type msinfo32 into the search box, and press Enter.
- In System Summary look for System Type. It can be either a x64-based PC (mean 64 bits) or a x86-based PC (mean 32 bits).

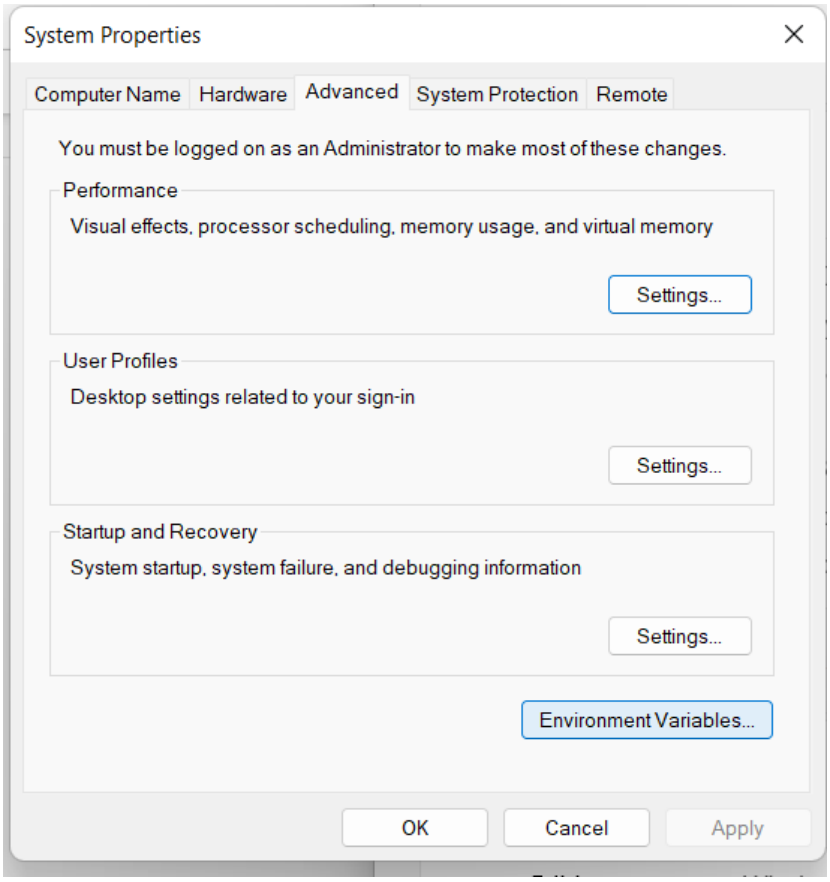
Co-funded by the Erasmus+ Programme of the European Union



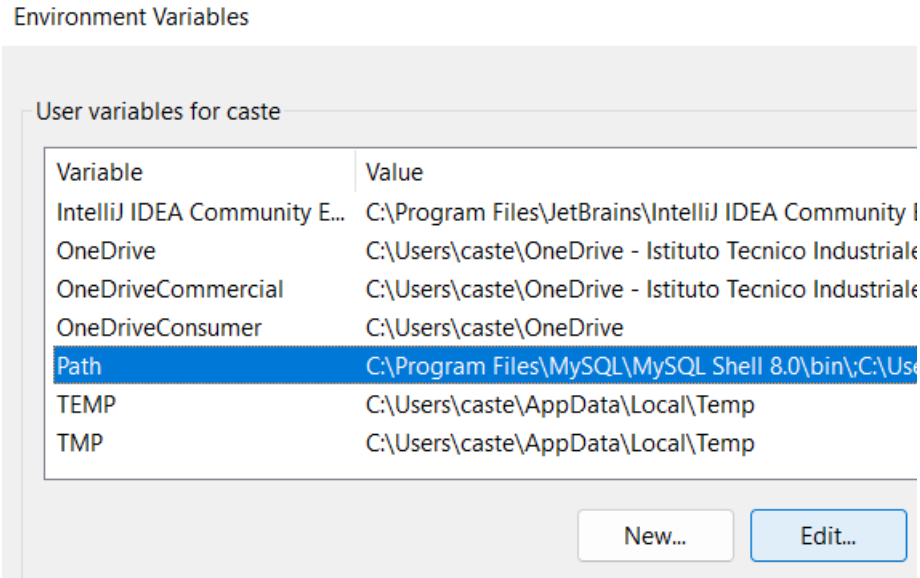
3. After the end of the installation, it is necessary to add the installation folder to the system path.
4. Open the System Information from Settings, and click on Advanced Settings



Co-funded by the Erasmus+ Programme of the European Union

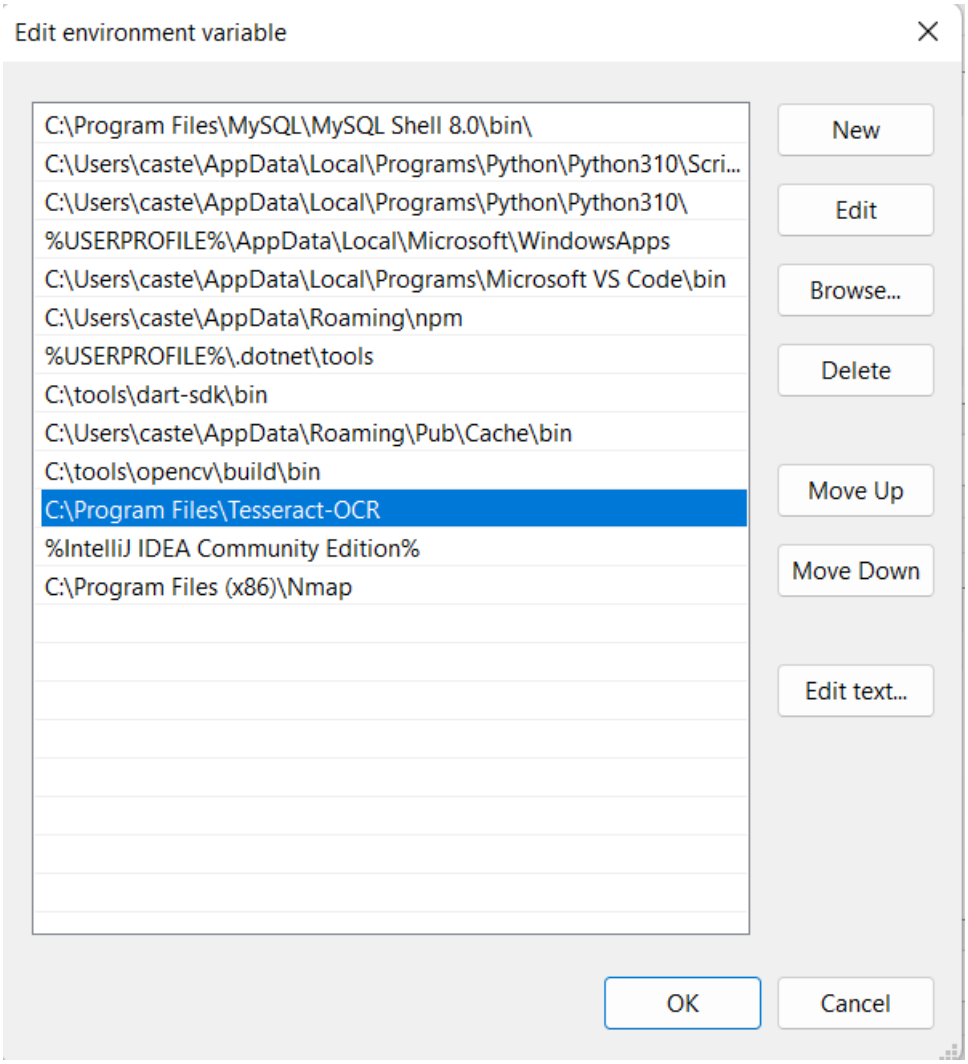


5. Select Environment Variables
6. Select the Path row and click edit



Co-funded by the
Erasmus+ Programme
of the European Union

7. Choose the installation path of Tesseract (typically c:\Program Files\Tesseract-OCR)



On Mac

Tesseract is available on Mac either via MacPorts or Homebrew.

If you use MacPorts run this command: `sudo port install tesseract`

If you use Homebrew run this command: `brew install tesseract`

On Linux

Tesseract is a available as a packeage for most of the linux distribution, so its installation is straightforward.
E.g. for installing Tesserect in Debian based distribution (like Ubuntu), run this command:

```
sudo apt install tesseract-ocr
sudo apt install libtesseract-dev
```

Co-funded by the Erasmus+ Programme of the European Union

Tesseract Python Library

In python it is available a module named **pytesseract** that is a wrapper for the Tesseract engine.
To install it in your environment, run this command: `pip install pytesseract`

Typically, it is not sufficient using only the pytesseract library, but in many case the images before to be sent to the OCR engine have to be pre-processes by applying resizing and filters.

The most used open-source image elaboration library is OpenCV and of course a dedicate module is available for python.

To install it in your environment, run this command: `pip install opencv-contrib-python`

OpenCV is used for opening image from files and for applying filters.

Using Tesseract in Python

A minimal python code for exploiting tesseract is the following:

```
1  import cv2
2  from pytesseract import*
3
4  custom_config = r'--oem 3 --psm 8'
5  image = cv2.imread('ocr2.jpeg')
6  gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
7  text = pytesseract.image_to_string(gray_image,config=custom_config,lang="ssd")
8  print(text)
```

- Lines 1 and 2 import the necessary modules: OpenCv (cv2) and Tesserect (pytesseract).
- Line 4 set a variable containing custom configuration parameters that will be better explained in the next section
- Line 5 exploits OpenCV for loading image ocr2.jpeg
- Line 6 converts the image to gray scale (typically tesseract works better with images in grayscale)
- Line 7 executes the function image_to_string of the tesseract module. This function returns the text read inside the picture passed as parameter. The function needs the following parameters:
  - Image: most of the time it is necessary applying some filters to the image
  - Custom configuration parameters: the next section will explain them
  - Language: it is an optional parameter, if it is not provided English language will be selected. It is also possible to select custom training models. In line 7 we select “ssd” as language model: it is one of the 3 we proposed as custom model for recognizing digits from lcd displays
- There is an alternative function to image_to_string, named image_to_data that returns a dictionary containing all the recognized text within the coordinates where are placed and the estimated accuracy.
- Line 8 print the output

Co-funded by the Erasmus+ Programme of the European Union

Configuration parameters

Tesseract engine supports two kind of configuration parameters: *page segmentation modes* and OCR modes.

There are 14 possible page segmentation modes that indicate how the text is assumed to be displayed inside the image given as input of the engine.
In order to set this parameter, it is necessary to use the --psm keyword before the number selected from the Table 1.

Segmentation Mode	Description
1	Orientation and script detection (OSD) only.
2	Automatic page segmentation with OSD.
3	Automatic page segmentation, but no OSD, or OCR. (not implemented)
4	Fully automatic page segmentation, but no OSD. (Default)
5	Assume a single column of text of variable sizes.
6	Assume a single uniform block of vertically aligned text.
7	Assume a single uniform block of text.
8	Treat the image as a single text line.
9	Treat the image as a single word.
10	Treat the image as a single word in a circle.
11	Treat the image as a single character.
12	Sparse text. Find as much text as possible in no particular order.
13	Sparse text with OSD.
14	Raw line. Treat the image as a single text line, bypassing hacks that are Tesseract-specific.

Table 1 Page Segmentation Modes (PSM)

There are 4 OCR engine modes that indicate how the Tesseract engine must operate. To exploit this configuration parameter, it is necessary to use the --oem keyword before the number reported in Table 2.

Co-funded by the Erasmus+ Programme of the European Union

Engine Modes	Description
1	Legacy engine only.
2	Neural nets LSTM engine only.
3	Legacy + LSTM engines.
4	Default, based on what is available.

Table 2 OCR Engine Modes

Projects

This section includes 4 projects, described step to step, that involve the Universal Measure Reader. Please note that all these projects are only proof of concept (POC), therefore they could not be ready to be used in real work environments. Before using these projects in real environments, it is necessary to consider the safety of the used electronic devices (design and develop containers that are compliant to IP65, IP66 or IP67 standards (12)) and the product engineering (e.g., moving from breadboard to printed circuits).
Some examples of 3D printer Arduino boxes are reported in (13).

Project 1: Capture a measure every 30 seconds

DIFFICULTY	Hardware  Software 
SOFTWARE REQUIREMENTS	Arduino IDE installed on Windows, Linux or macOS
HARDWARE REQUIREMENTS	ESP32-CAM board Programmer (e.g., ESP32-CAM-MB), only for uploading the sketch
TIME	about 40 minutes

Co-funded by the Erasmus+ Programme of the European Union

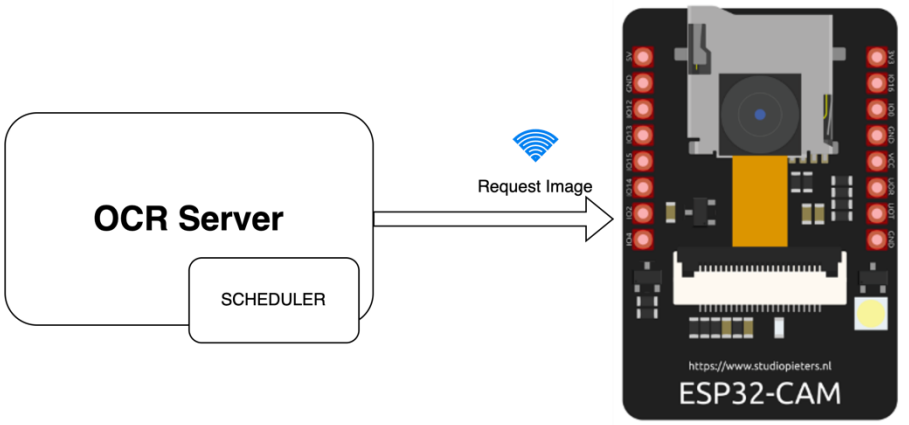


Figure 14 Project 1 overview

1. As first step is necessary to setup the Arduino IDE for the ESP32-CAM board as described in Tutorial 1: Adding the ESP32 library to Arduino IDE
2. The sample code provided within the **espressif** library is already fully working and it provides a web service that allow to control the camera. Nevertheless, we suggest enhancing such web service by adding two methods (led-on and led-off) for controlling also the powerful white led included in the board. In according with the old device display it could be necessary to modify the light conditions for better recognize the measure.

   It is necessary to modify the file app_httpd.cpp by
3. adding the following static functions:

```
347
348  static esp_err_t ledon_handler(httpd_req_t *req)
349  {
350    esp_err_t res = ESP_OK;
351
352    digitalWrite(4, HIGH);
353
354    return res;
355
356  }
357
358  static esp_err_t ledoff_handler(httpd_req_t *req)
359  {
360    esp_err_t res = ESP_OK;
361    digitalWrite(4, LOW);
362
363    return res;
364
365  }
```

Co-funded by the Erasmus+ Programme of the European Union

These functions basically set HIGH or LOW the value of the LED (GPIO 4); they receive and http request as parameter even if they don't use it and return and esp_err_t object that can assume the values ESP_OK or ESP_FAIL in according if the operation is correctly performed.

4. Define the uri registration of the two functions

```
1250    //romuas adding led management
1251    httpd_uri_t ledon_uri = {
1252        .uri = "/led-on",
1253        .method = HTTP_GET,
1254        .handler = ledon_handler,
1255        .user_ctx = NULL};
1256
1257    httpd_uri_t ledoff_uri = {
1258        .uri = "/led-off",
1259        .method = HTTP_GET,
1260        .handler = ledoff_handler,
1261        .user_ctx = NULL};
1262
```

httpd_uri_t structure allows to define a new http handler for the web service. It is necessary to define the uri, the http method and the function handler that will be called by the web service.

5. Adding the function to the web service

```
1272    if (httpd_start(&camera_httpd, &config) == ESP_OK)
1273    {
1274        httpd_register_uri_handler(camera_httpd, &index_uri);
1275        httpd_register_uri_handler(camera_httpd, &cmd_uri);
1276        httpd_register_uri_handler(camera_httpd, &status_uri);
1277        httpd_register_uri_handler(camera_httpd, &capture_uri);
1278        httpd_register_uri_handler(camera_httpd, &bmp_uri);
1279
1280        httpd_register_uri_handler(camera_httpd, &xclk_uri);
1281        httpd_register_uri_handler(camera_httpd, &reg_uri);
1282        httpd_register_uri_handler(camera_httpd, &greg_uri);
1283        httpd_register_uri_handler(camera_httpd, &pll_uri);
1284        httpd_register_uri_handler(camera_httpd, &win_uri);
1285        httpd_register_uri_handler(camera_httpd, &ledon_uri);
1286        httpd_register_uri_handler(camera_httpd, &ledoff_uri);
1287    }
1288
```

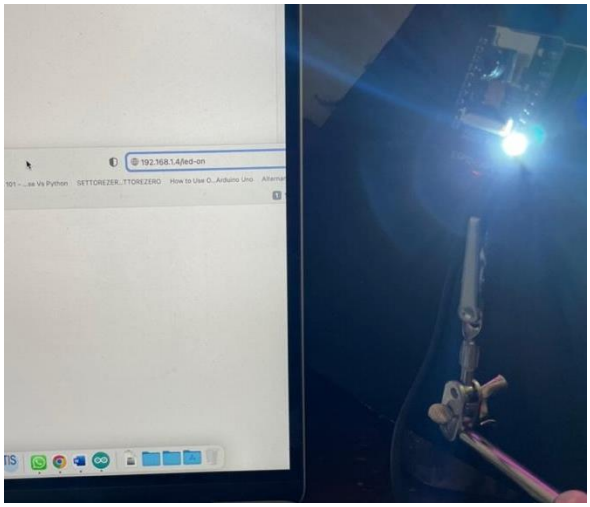
The highlighted lines register the above defined httpd_uri_t objects to the web service.

Co-funded by the Erasmus+ Programme of the European Union

6. Upload the modify code to the ESP32-CAM board.

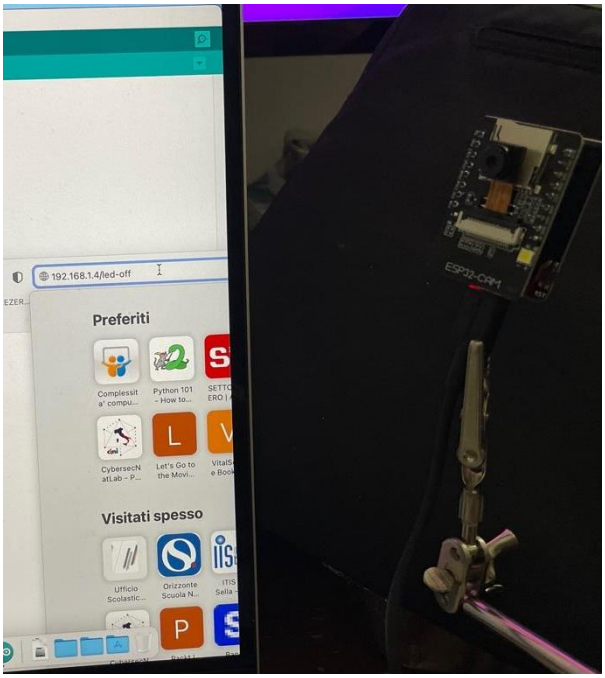
After uploading the sketch, you can see in the serial monitor the ip address assigned by your router to the ESP32-CAM. In our case is 192.168.1.4.

In order to switch on the led you need to open a browser an insert the the following address:
http://{ESP32-CAM IP}/led-on
In our case http://192.168.1.4/led-on



In order to switch off the led you need to open a browser an insert the the following address:
http://{ESP32-CAM IP}/led-off
In our case http://192.168.1.4/led-off

Co-funded by the Erasmus+ Programme of the European Union



7. After finishing the Arduino coding, it is necessary to move on the OCR server part. We need to develop a program that
8. every X seconds performs a request to the ESP32-CAM web service for retrieving a picture
9. then it exploits the tesseract engine for recognize the measure
10. and at last displaying the measure
11. We start coding the OCR server scheduler in python by installing two libraries: schedule and requests:

```
pip install requests
pip install schedule
```

Requests library is necessary for performing HTTP requests towards the ESP32-CAM web server, while the schedule library offers functionality for scheduling the execution of tasks.
As described in the Software section we will make use of OpenCV and Tesseract libraries and also of our ocr library that wraps the Tesseract engine.

```
1   import schedule
2   import requests
3   import time
4   import json
5   import shutil
6   from ocr import ocr_image
7
```

Co-funded by the Erasmus+ Programme of the European Union

Lines 1 to 6 import the required modules:

- schedule: it allows to schedule tasks
- requests: it allows to perform http requests
- time: it is used for the sleep function in the main loop of the program
- json: it is used for retrieving the configuration stored in a json file
- shutil: it is used for copying the download file from the ESP32-CAM
- ocr: it is our wrapper of the Tesseract engine

12. Loading configuration

The application configuration parameters should be stored outside our source code in order to allow to easily modify them. Json is one of the most common file format for storing configurations and thanks to the json library it is very easy to manage such file.

Figure 15 shows an example of configuration file for our scheduler application.

The schedule_confing.json file allows to define:

- the image capture interval expressed in seconds (capture_seconds)
- the ESP32-CAM web server url (capture_url)
- the Tesseract page segmentation mode (psm)
- the Tesseract language (dictionary)
- the crop zone if any (crop, crop-data)
- if it is required to show a debug window (showWindow)

schedule_config.json > ...

```
1   {
2       "caputure_seconds": 1,
3       "capture_url": "http://192.168.1.4/capture",
4       "psm": 9,
5       "dictionary": "ssd",
6       "crop": true,
7       "crop-data": { "x": 380, "y": 490, "width": 210, "height": 320 },
8       "showWindow": true
9
10  }
```

Figure 15 Example of scheduler configuration

Co-funded by the Erasmus+ Programme of the European Union

Loading a json file in Python is quite easy as reported in the below code:

```
 8    def loadConfiguration():
 9        print('Loading configuration')
10        with open('schedule_config.json') as config_file:
11            configuration = json.load(config_file)
12            return configuration
```

This function opens the configuration file and copy its content in a dictionary.
The configuration in Figure 15 is converted in the below python object:

```
{'caputure_seconds': 1, 'capture_url': 'http://192.168.1.4/capture', 'psm': 9, 'dictionary': 'ssd', 'crop': True, 'crop-data':
{'x': 380, 'y': 490, 'width': 210, 'height': 320}, 'showWindow': True}
```

13. Capture the measure

Capturing the measure involves several operations as reported in the code below.

```
14  def caputureImage(configuration):
15      global index
16      image_url=configuration['capture_url']
17      filename = image_url.split("/")[-1]
18      r = requests.get(image_url, stream = True)
19
20      if r.status_code == 200:
21          r.raw.decode_content = True
22          filename = str(index)+"-"+filename+".jpeg"
23          with open(filename,'wb') as f:
24              shutil.copyfileobj(r.raw, f)
25
26          index+=1
27          print('Image sucessfully Downloaded: ',filename)
28          text, conf = ocr_image(filename,configuration['psm'],
29                                 configuration['dictionary'],
30                                 configuration['showWindow'],
31                                 crop=configuration["crop-data"].values()
32                                 if configuration["crop"] else None )
33          print("Found text", text,conf)
34      else:
35          print('Image Couldn\'t be retreived')
36
```

- Line 14 defines the captureImage function that has parameter the configuration read from the json file
- Line 15 retrieves the global variable named index and used for storing the image with different names. It is increased after every capture.
- Line 16 retrieves the url of the web service from the configuration object
- Line 17 performs an HT<PP GET request for retrieving the image from the ESP32-CAM

Co-funded by the Erasmus+ Programme of the European Union

- Line 20 checks the requests status: 200[5] means that the request was OK
- Line 21 indicates that the raw content of the response must be decoded in according with the Content-Type http header[6]
- Line 22 generates an unique file name
- Lines 23 and 24 save the response content into a image file
- Lines from 28 to 32 calls the ocr_image function passing all the configuration parameters
- Line 33 displays the recognized text, while Line 35 reports the occurred errors (if any)

14. Schedule the image capture in the main function

```
37  if __name__ == "__main__":
38      index = 1
39      configuration = loadConfiguration()
40      capture_seconds = configuration['caputure_seconds']
41      schedule.every(capture_seconds).seconds.do(caputureImage, configuration)
42      caputureImage(configuration)
43      while True:
44          schedule.run_pending()
45          time.sleep(1)
```

- Line 37 checks if the script has run directly and has not been included as module
- Line 38 declares a global variable used for generating unique image names
- Line 39 loads the scheduler configuration
- Line 40 retrieves the capture time interval
- Line 41 configures the schedule object for executing the captureImage function in according with the time interval
- Line 42 execute the captureImage function for the first time without waiting the capture time interval
- Lines from 43 to 45 represents the main loop of the scheduler. The run_pending method of schedule object has to be executed continuously in order to enable the scheduler to execute the task in the proper time.

The time.sleep function is necessary to allow other threads to take the control of the system for a while.

[5] More information about the HTTP statis code is available at https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

[6] More information about the Content-Type header is available at https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Type

Co-funded by the Erasmus+ Programme of the European Union

15. Results

As example we tried to capture the temperature measure from a domestic meteorological station. We started from the raw captured image (Figure 16) then through the configuration we indicate to crop the image in a specific zone (We are interested in the internal temperature only, that is shown in the left lower corner of the display).
Figure 17 shows the output of the program: it is displayed the cropped image with the result of the Tesseract engine recognition.

Figure 16 Raw Image

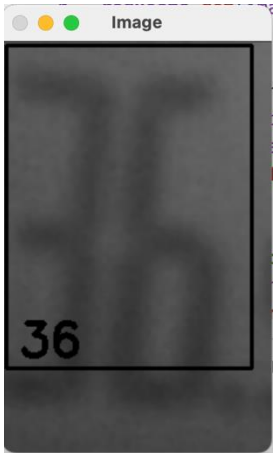


Figure 17 Cropped Image with Tesseract Recognition

Co-funded by the
Erasmus+ Programme
of the European Union

Project 2: Capture a measure on a button press

DIFFICULTY	Hardware Easy Medium Hard  Software Easy **Medium** Hard 
SOFTWARE REQUIREMENTS	Arduino IDE installed on Windows, Linux or macOS, Tesseract Engine
HARDWARE REQUIREMENTS	ESP32-CAM board Programmer (e.g., ESP32-CAM-MB), only for uploading the sketch Breadboard Breadboard power supply 2 resistors (10 KΩ) 2 push buttons Wires
TIME	about 60 minutes

The ESP32-CAM board is programmed for sending an image to our OCR server when a button is pressed. For convenience the bright white led is controlled by another button. The OCR server provides a web service for uploading the image and exploiting Tesseract Engine for recognizing the measures.
The OCR server also provides a web page that shows the retrieved measures. A high-level overview of this project is depicted in Figure 18.

Co-funded by the Erasmus+ Programme of the European Union

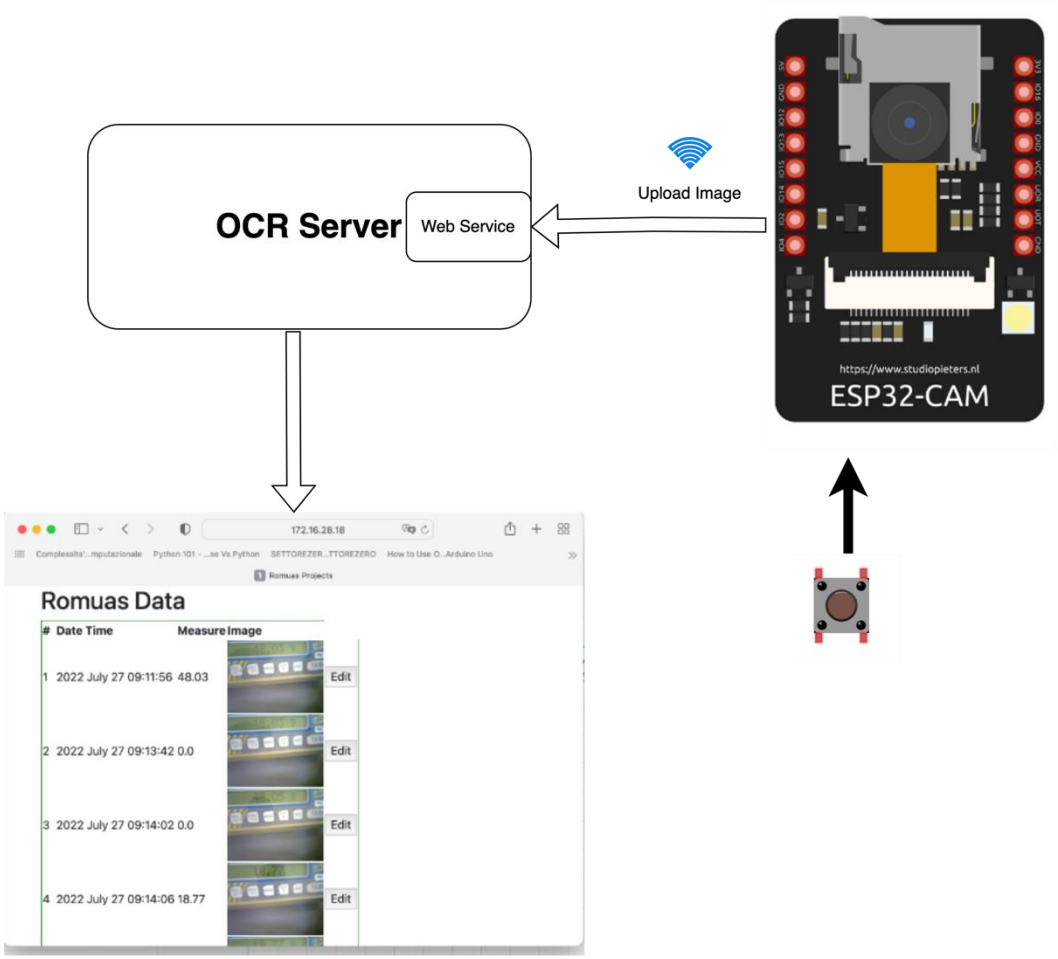


Figure 18 Project 2 high level overview

1. As first step it is necessary to wire the component as shown in Figure 19.
    - Connect the power supply to the breadboard for providing 5V and Ground to the Power rails
    - Connect the ESP32-CM 5V pin and GND to the respective Power Trailing lines
    - Connect the GPIO 14 to the button A and with a 10 K resistance to the GND
    - Connect the GPIO 2 to the button B and with a 10 K resistance to the GND
    - Provide 5V to the buttons

Co-funded by the Erasmus+ Programme of the European Union

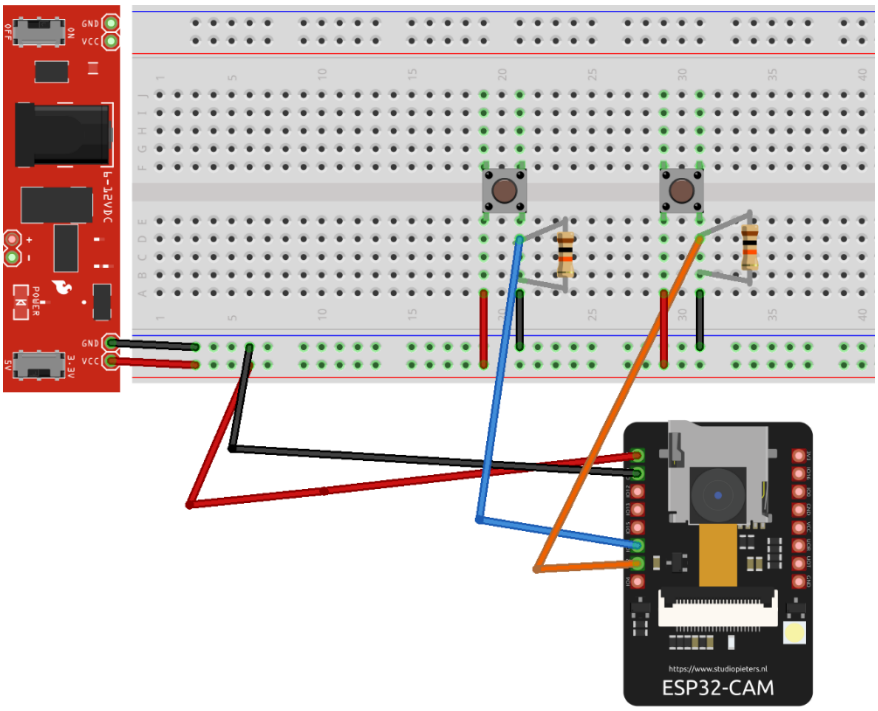


Figure 19 Project 2 Wiring

2. Create a new Arduino Sketch.

   The proposed sketch is mostly based on the work presented in (14), with the main difference about the server side of the solution: we will make us of simple python web server while the article proposes PHP.

3. Import the needed libraries.

```
13    #include <Arduino.h>
14    #include <WiFi.h>
15    #include "soc/soc.h"
16    #include "soc/rtc_cntl_reg.h"
17    #include "esp_camera.h"
```

   - Line 13 includes the standard Arduino library, used for functions like digitalWrite, digitalRead, pinMode, etc.
   - Line 14 includes the WiFi library used for connecting the ESP32-CAM board to a Wireless network and creating an HTTP client for sending the picture to the OCR server
   - Lines from 15 to 17 includes the specific libraries for the ESP32-CAM board

4. Define the configuration variables

Co-funded by the Erasmus+ Programme of the European Union

```
18
19  const char* ssid =
20  const char* password =
21
22  String serverName = "172.16.28.18";
23
24  String serverPath = "/upload-image";     // The server path for uploading the IMAG
25
26  const int serverPort = 5001;
27
28  const int buttonPhotoPin = 2;  //change with the PIN you've connected the button
29  const int buttonFlashPin = 14;
30  int buttonPhotoState = 0;
31  int buttonFlashState = LOW;
32  const int ledPin = 4;
33
34  WiFiClient client;
```

- Lines 19 and 20 allow to set the Wi-Fi network connection parameters: the SSID (name of the wireless network) and the password.
- Lines from 22 to 26 allow to set the address of the web service where uploading the images. The address is composed by the serverName (or ip address), the port and the server path. In our example the final address will be http://172.16.28.18:5001/upload-image
- Lines from 28 to 32 define the pin where the buttons are connected and their states
- Line 34 declares the WiFiClient used for sending the picture

5. Define the GPIOs camera configurations

```
36  // CAMERA_MODEL_AI_THINKER
37  #define PWDN_GPIO_NUM     32
38  #define RESET_GPIO_NUM    -1
39  #define XCLK_GPIO_NUM      0
40  #define SIOD_GPIO_NUM     26
41  #define SIOC_GPIO_NUM     27
42
43  #define Y9_GPIO_NUM       35
44  #define Y8_GPIO_NUM       34
45  #define Y7_GPIO_NUM       39
46  #define Y6_GPIO_NUM       36
47  #define Y5_GPIO_NUM       21
48  #define Y4_GPIO_NUM       19
49  #define Y3_GPIO_NUM       18
50  #define Y2_GPIO_NUM        5
51  #define VSYNC_GPIO_NUM    25
52  #define HREF_GPIO_NUM     23
53  #define PCLK_GPIO_NUM     22
```

Co-funded by the Erasmus+ Programme of the European Union

6. Define a function for configuring the camera

```
camera_config_t configCamera(){
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;
  // init with high specs to pre-allocate larger buffers
  if(psramFound()){
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 10;  //0-63 lower number means higher quality
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_CIF;
    config.jpeg_quality = 12;  //0-63 lower number means higher quality
    config.fb_count = 1;
  }
  return config;
}
```

The camera_config_t object is initialized with the pin configuration

7. Define a function for starting the camera

```
void startCamera(camera_config_t config){
  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    delay(1000);
    ESP.restart();
  }


}
```

Co-funded by the
Erasmus+ Programme
of the European Union

8. Define the function for sending the photo

```
104 String sendPhoto() {
105   String getAll;
106   String getBody;
107
108   camera_fb_t * fb = NULL;
109   fb = esp_camera_fb_get();
110   if(!fb) {
111     Serial.println("Camera capture failed");
112     delay(1000);
113     ESP.restart();
114   }
115
116   Serial.println("Connecting to server: " + serverName);
117
118   if (client.connect(serverName.c_str(), serverPort)) {
119     Serial.println("Connection successful!");
120     String head = "--Romuas\r\nContent-Disposition: form-data; name=\"file\"; filename=\"esp32-cam.jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";
121     String tail = "\r\n--Romuas--\r\n";
122
123
124     uint32_t imageLen = fb->len;
125     uint32_t extraLen = head.length() + tail.length();
126     uint32_t totalLen = imageLen + extraLen;
127
128     client.println("POST " + serverPath + " HTTP/1.1");
129     client.println("Host: " + serverName);
130     client.println("Content-Length: " + String(totalLen));
131     client.println("Content-Type: multipart/form-data; boundary=Romuas");
132     client.println();
133     client.print(head);
134
135     uint8_t *fbBuf = fb->buf;
136     size_t fbLen = fb->len;
137     for (size_t n=0; n<fbLen; n=n+1024) {
138       if (n+1024 < fbLen) {
139         client.write(fbBuf, 1024);
140         fbBuf += 1024;
141       }
142       else if (fbLen%1024>0) {
143         size_t remainder = fbLen%1024;
144         client.write(fbBuf, remainder);
145       }
146     }
147     client.print(tail);
148
149     esp_camera_fb_return(fb);
150
151     int timoutTimer = 10000;
152     long startTimer = millis();
153     boolean state = false;
154
155     while ((startTimer + timoutTimer) > millis()) {
156       Serial.print(".");
157       delay(100);
158       while (client.available()) {
159         char c = client.read();
160         if (c == '\n') {
161           if (getAll.length()==0) { state=true; }
162           getAll = "";
163         }
164         else if (c != '\r') { getAll += String(c); }
165         if (state==true) { getBody += String(c); }
166         startTimer = millis();
167       }
168       if (getBody.length()>0) { break; }
169     }
170     Serial.println();
171     client.stop();
172     Serial.println(getBody);
173   }
174   else {
175     getBody = "Connection to " + serverName +  " failed.";
176     Serial.println(getBody);
177   }
178   return getBody;
179 }
```

Co-funded by the
Erasmus+ Programme
of the European Union

9. Define the setup function that is called once at the start of the program

```
void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);

  pinMode(buttonPhotoPin, INPUT);
  pinMode(buttonFlashPin, INPUT);
  pinMode(ledPin, OUTPUT);

  Serial.begin(115200);

  WiFi.mode(WIFI_STA);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("ESP32-CAM IP Address: ");
  Serial.println(WiFi.localIP());

  camera_config_t config =configCamera();
  startCamera(config);

  //sendPhoto();
}
```

10. Define the main loop of the program

```
void loop() {
  buttonPhotoState = digitalRead(buttonPhotoPin);
  int currentButtonFlashState = digitalRead(buttonFlashPin);
  if (currentButtonFlashState == HIGH){
    buttonFlashState = buttonFlashState == HIGH ? LOW : HIGH;
    digitalWrite(ledPin, buttonFlashState);
  }
  Serial.println(buttonPhotoState);
  if (buttonPhotoState == HIGH) {
   sendPhoto();
  }
}
```

11. Now, it is the turn of the OCR web server implementation

We decided to use Flask framework[7] for implementing a web server that provide both a method for uploading the image and recognize the measure and a web page for displaying the results.

First, we need to install two further libraries:

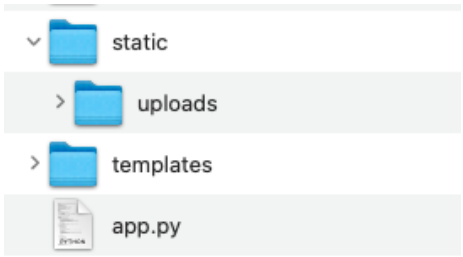
```
pip install flask
pip install xlswriter
```

[7] You can find more information at the official website https://flask.palletsprojects.com/en/2.1.x/

Co-funded by the Erasmus+ Programme of the European Union

12. Create a Flask Project

The project has the following structure:



The static/uploads folder stores the uploaded images, the templates folder contains the html page templates, while the app.py file contains the code of the application

13. Import the required libraries

```
1  import os
2  from flask import Flask, jsonify, render_template, request, send_file
3  from ocr import romuas_ocr, ocr_core
4  from datetime import datetime
5  import xlsxwriter
```

Line 1 imports the os library used for performing operation on upload files

Line 2 imports the flask functions

Line 3 imports our ocr library

Line 5 imports the library for exporting data in Excel format

14. Initialize the Flask application

```
8  # define a folder to store and later serve the images
9  UPLOAD_FOLDER = 'static/uploads/'
10
11 # allow files of a specific type
12 ALLOWED_EXTENSIONS = set(['png', 'jpg', 'jpeg'])
13
14 samples = []
15 img_number = 0
16
17 app = Flask(__name__)
18 app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
```

Lines from 8 to 18 declare some config parameters, i.e. the upload folder, the allowed image extension and create the flask app.

The sample variable will store the measures, while img_number is used for generating unique name of the uploaded pictures.

Co-funded by the Erasmus+ Programme of the European Union

15. Define a function for checking the uploaded file

```
20  # function to check the file extension
21  def allowed_file(filename):
22      return '.' in filename and \
23             filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
24
```

The function checks if the file extension is in the list of allowed extensions.

16. Create routing for the home page

```
26  @app.route('/')
27  def home_page():
28      return render_template('index.html')
29
```

In Flask the routing (i.e. , the mapping between address and executed method) is defined via decorator[8]. The lines from 26 to 28 assign to the route '/' the function home_page that simply displays the index.html web page

17. Define a function for exporting and Excel file with the recognized measures

```
30  @app.route('/download-excel',methods=['GET'])
31  def dowload_excel():
32      with xlsxwriter.Workbook('data-tmp-2.xlsx') as workbook:
33          worksheet = workbook.add_worksheet()
34          worksheet.write_row(0, 0, ('#','Date Time','Measure'))
35          for row_num, data in enumerate(map(lambda s: (s[3],s[0],s[1]) ,samples)):
36              worksheet.write_row(row_num+1, 0, data)
37          workbook.close()
38      try:
39          return send_file('data-tmp-2.xlsx', download_name='excel_export.xlsx')
40      except Exception as e:
41          return str(e)
```

In this function the xlsxwriter library is exploited for generating an Excel files containing the gathered measures.

[8] Python decorators were firstly defined in https://peps.python.org/pep-0318/

18. Define the function for managing the upload of the pictures

```
43  @app.route('/upload-image',methods=['POST'])
44  def uploadImage():
45      if 'file' not in request.files:
46          return "No file"
47      file = request.files['file']
48          # if no file is selected
49
50
51      if file and allowed_file(file.filename):
52          global img_number
53          filePath = os.path.join(app.config['UPLOAD_FOLDER'],str(img_number)+"-"+file.filename)
54          img_number+=1
55          file.save(filePath)
56          print("Saved",filePath)
57          ocr = romuas_ocr(filePath)
58          text = ocr.show_ocr((560,215,230,120))
59          samples.append((datetime.now().strftime("%Y %B %d %H:%M:%S"),text,filePath,img_number))
60          return "OK"
```

This function checks if is present a file in the request and if it has a valid extension, then the file is saved on the server.

After the file is saved, the Tesseract engine is executed (lines 57 and 58) and the result is added to the samples object that stores the recognized measure, the image and the time reference (line 59).

19. Define the routing for the results page

```
62  @app.route('/samples', methods=['GET'])
63  def samples_page():
64      return render_template('samples.html', data=samples)
```

When the user try to access to the /samples url, the samples.html template will be displayed having as parameter the object containing the measures.

20. Executing the application

```
if __name__ == '__main__':
    app.run(host='0.0.0.0',port=5001)
```

This is the code for starting the web server. The host parameter ‘0.0.0.0’ allows to publish the web server to the network, while the port parameter allows to define the port.

21. Define the sample page template

The core of the sample template is displaying the measure, this is implemented by the following lines

Co-funded by the Erasmus+ Programme of the European Union

```
<tbody>
        {% for sample in data %}
         <tr>
           <td> {{sample[3]}} </td>
           <td> {{sample[0]}} </td>
           <td> <span id="mes-{{sample[3]}}">{{sample[1]}}</span>
             <input style="display:none" name="edit-{{sample[3]}}" id="edit-{{sample[3]}}" value="{{sample[1]}}" </td>
           <td> <img src="{{sample[2]}}" height=100 /> </td>
           <td> <button name="edit" id="editb-{{sample[3]}}"
                value="{{sample[3]}}">Edit </button>
              <button name="save" id="save-{{sample[3]}}" value="{{sample[3]}}" style="display:none">Save </button>
              <button name="cancel" id="cancel-{{sample[3]}}" value="{{sample[3]}}" style="display:none">Cancel </button>
           </td>
         </tr>
        {% endfor %}
    </tbody>
    </table>
    {% endif %}
    <a href="/download-excel" download="Export-Excel.xlsx">Export in Excel</a>
```

Basically, it executes an iteration on all the measures (samples object) and create a row in a table.

22. Conclusion

This project has been exploited in two experiments described in "RIFERIMENTO Documento Esperimenti" (see Figure 20)

Figure 20 Project Results: left an example of setup, right the Sample web page reporting the captured measures

Project 3: Switch on a green led on a *correct* measure

DIFFICULTY	Hardware Easy Medium Hard  Software Easy **Medium** Hard 
SOFTWARE REQUIREMENTS	Arduino IDE installed on Windows, Linux or macOS, Tesseract Engine
HARDWARE REQUIREMENTS	Project 2 requirements and in addition: • Arduino Nano 33 IOT[9] • RGB led • 3 resistors (220Ω)
TIME	about 30 minutes if the projects 2 has already been completed

This project is an extension of Project 2: Capture a measure on a button press. The OCR web service is modified for managing a threshold on the recognized measures and if the measure is in the specified valid interval a message to an Arduino Nano IOT board is sent for change the color to green of an RGB led, and as opposite when the measure is out of the valid range the led color is changed to red.
In Figure 21 is shown a high-level overview of this project.

[9] As alternative, you can use other boards including WI-FI connectivity

Co-funded by the Erasmus+ Programme of the European Union

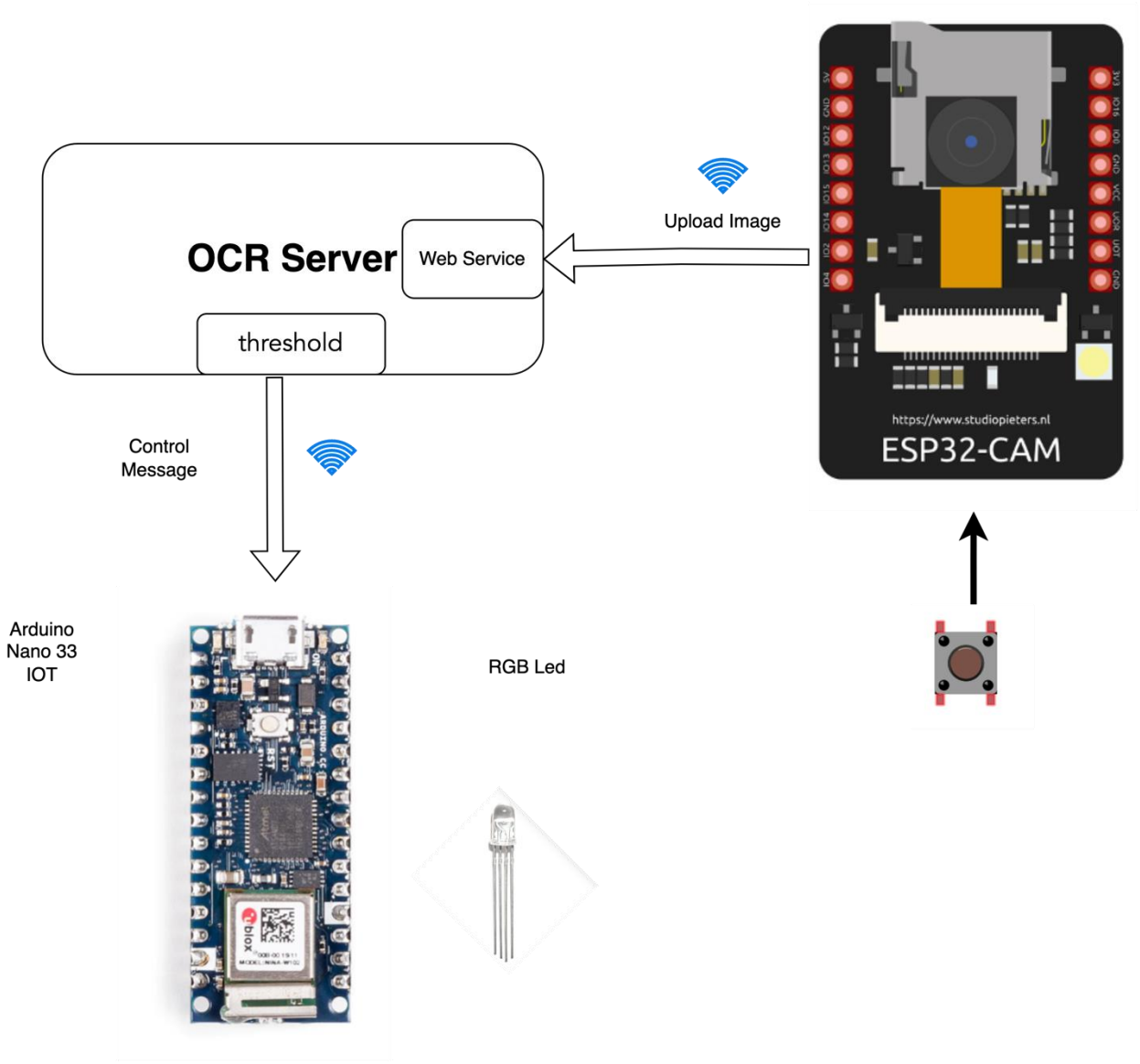


Figure 21 Project 3 high-level overview

1. As first step of the project it is necessary to do the wiring as depicted in Figure 22 and Figure 23:
    - Connecting the Arduino Nano 33 IOT D2 pin to the first leg (RED) of the RGB led
    - Connecting the Arduino Nano 33 IOT D3 pin to the third leg (GREEN) of the RGB led
    - Connecting the Arduino Nano 33 IOT D5 pin to the fourth leg (BLUE) of the RGB led
    - Connecting the Arduino Nano 33 IOT GND pin to the second leg (cathod) of the RGB led

Co-funded by the
Erasmus+ Programme
of the European Union

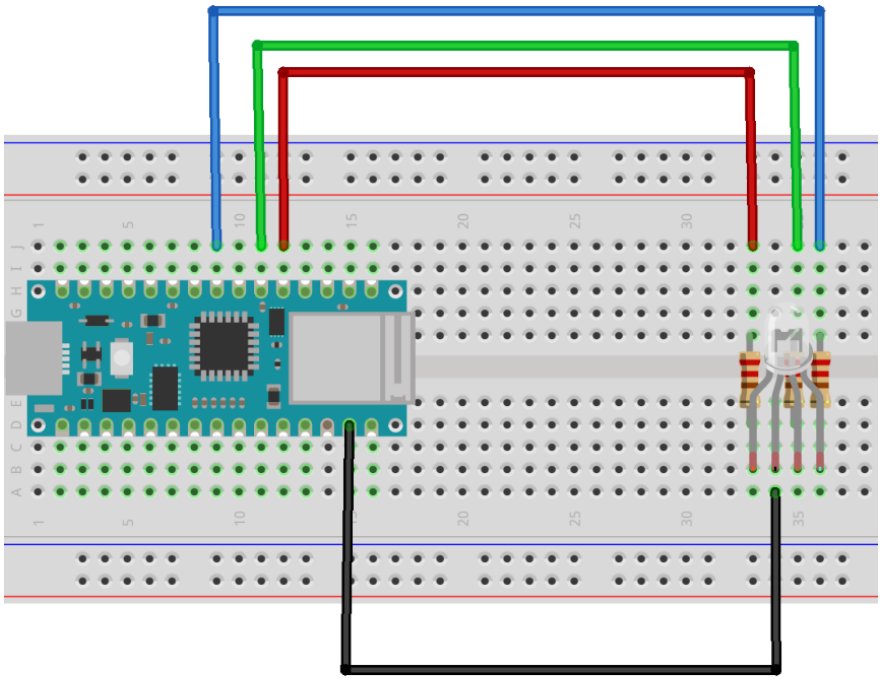


Figure 22 Project 3 - Breadboard View

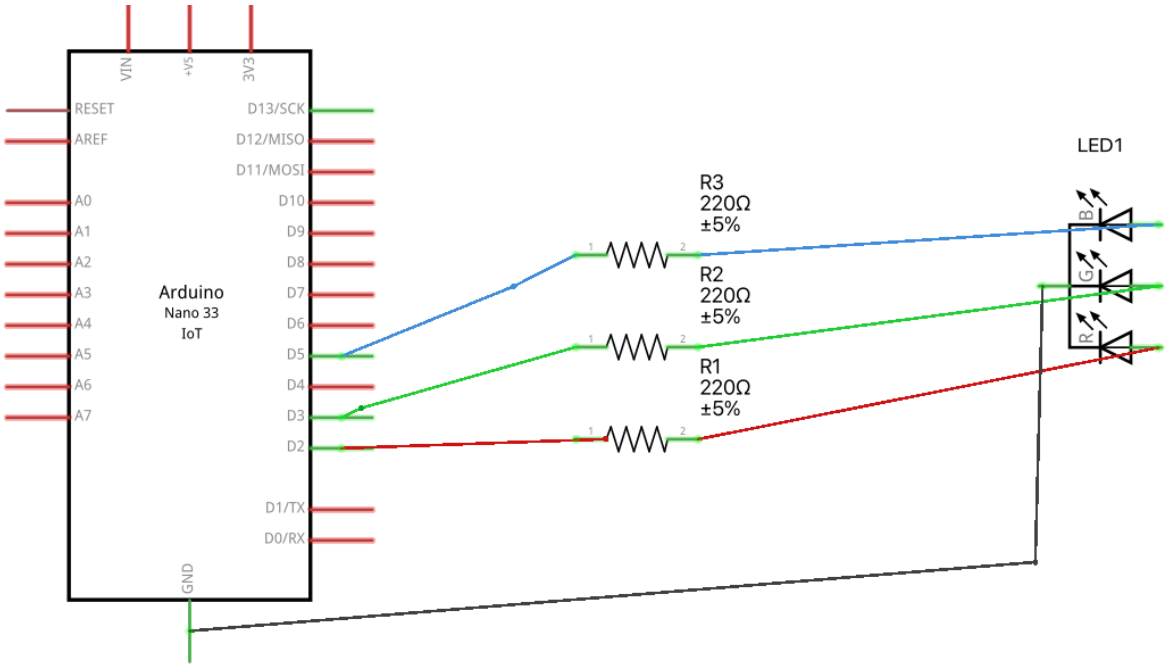
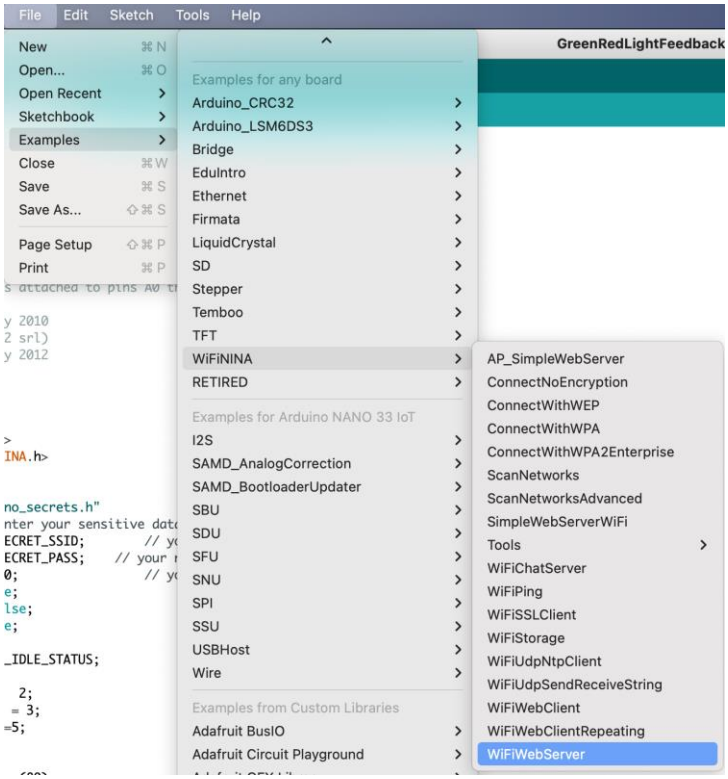


Figure 23 Project 3 Logic View

Co-funded by the
Erasmus+ Programme
of the European Union

2. Write the sketch.

   We suggest starting from the WiFiNINA / WiFiWebServer example



3. Modify the arduino_secrets.h with SSID and Password of your Wi-Fi connection

Co-funded by the
Erasmus+ Programme
of the European Union

4. Define the variables for storing the network parameter, the pin mapping, the connection status and the server.

```
/*
  WiFi Web Server

 A simple web server that changes to color of a RGB led connected to D2 (R), D3 (G) and D5 (B)

 */

#include <SPI.h>
#include <WiFiNINA.h>


#include "arduino_secrets.h"
///////please enter your sensitive data in the Secret tab/arduino_secrets.h
char ssid[] = SECRET_SSID;        // your network SSID (name)
char pass[] = SECRET_PASS;    // your network password (use for WPA, or use as key for WEP)
int keyIndex = 0;                 // your network key index number (needed only for WEP)


int status = WL_IDLE_STATUS;

const int RED =  2;   //pin D2
const int GREEN = 3;   //pin D3
const int BLUE =5;    //pin D5


WiFiServer server(80);
```

5. Define the functions for change the led color to Green, to Red or to switch it off

```
void lightOff(){
  digitalWrite(RED, LOW);
  digitalWrite(GREEN, LOW);
  digitalWrite(BLUE, LOW);
  Serial.println("OFF");

}

void lightRed(){
  digitalWrite(RED, HIGH);
  digitalWrite(GREEN, LOW);
  digitalWrite(BLUE, LOW);
  Serial.println("light Red");

}

void lightGreen(){
  digitalWrite(RED, LOW);
  digitalWrite(GREEN, HIGH);
  digitalWrite(BLUE, LOW);
  Serial.println("light Green");

}
```

Co-funded by the Erasmus+ Programme of the European Union

6. Define the setup function, that is executed once at the application start (or reset)

This function

- initializes the pin (lines from 54 to 57) and the serial connection (lines 58 to 61). Using the serial connection is optional, you can comment such lines. We use serial connection to retrieve information about the IP assigned to the Arduino board.
- Lines from 64 to 68 check the status of the WIFI module.
- Lines from 70 to 63 check if there is a firmware upgrade available[10]
- Lines from 76 to 85 perform the WIFI connection and wait until the board is connected
- At last in line 87 the details of the WIFI connection are printed

```
53 void setup() {
54   pinMode(RED, OUTPUT);
55   pinMode(GREEN, OUTPUT);
56   pinMode(BLUE, OUTPUT);
57   //Initialize serial and wait for port to open:
58   Serial.begin(9600);
59   while (!Serial) {
60     ; // wait for serial port to connect. Needed for native USB port only
61   }
62
63   // check for the WiFi module:
64   if (WiFi.status() == WL_NO_MODULE) {
65     Serial.println("Communication with WiFi module failed!");
66     // don't continue
67     while (true);
68   }
69
70   String fv = WiFi.firmwareVersion();
71   if (fv < WIFI_FIRMWARE_LATEST_VERSION) {
72     Serial.println("Please upgrade the firmware");
73   }
74
75   // attempt to connect to WiFi network:
76   while (status != WL_CONNECTED) {
77     Serial.print("Attempting to connect to SSID: ");
78     Serial.println(ssid);
79     // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
80     status = WiFi.begin(ssid, pass);
81
82     // wait 10 seconds for connection:
83     delay(10000);
84   }
85   server.begin();
86   // you're connected now, so print out the status:
87   printWifiStatus();
88
89 }
90
```

[10] More information about how to update the Arduino Nano 33 Iot firmware is available at https://support.arduino.cc/hc/en-us/articles/360013896579-Check-and-update-the-firmware-for-WiFiNINA-and-WiFi101

Co-funded by the Erasmus+ Programme of the European Union

7. Define the print WiFi status function

```
void printWifiStatus() {
  // print the SSID of the network you're attached to:
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());

  // print your board's IP address:
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);

  // print the received signal strength:
  long rssi = WiFi.RSSI();
  Serial.print("signal strength (RSSI):");
  Serial.print(rssi);
  Serial.println(" dBm");
}
```

8. Define the main loop.

   The loop function manage the client connection, whenever a client is connected the request is analyzed in order to understand if it request to change the color of the RGB led or to switch off the led.

   The clients must compose the URL as follow:

   http://{Arduino board assigned id}/green change the RGB led color to green

   http://{Arduino board assigned id}/red change the RGB led color to red

   http://{Arduino board assigned id}/off switch off the led

   The url parsing is perfomed from line 153 to 162.

   The server returns in any case a web page with Romuas Client text.

Co-funded by the Erasmus+ Programme of the European Union

```
void loop() {

 // listen for incoming clients
 WiFiClient client = server.available();   // listen for incoming clients

  if (client) {                             // if you get a client,

    Serial.println("new client");           // print a message out the serial port

    String currentLine = "";                // make a String to hold incoming data from the client

    while (client.connected()) {            // loop while the client's connected

      if (client.available()) {             // if there's bytes to read from the client,

        char c = client.read();             // read a byte, then

        Serial.write(c);                    // print it out the serial monitor

        if (c == '\n') {                    // if the byte is a newline character

          // if the current line is blank, you got two newline characters in a row.

          // that's the end of the client HTTP request, so send a response:

          if (currentLine.length() == 0) {

            // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)

            // and a content-type so the client knows what's coming, then a blank line:

            client.println("HTTP/1.1 200 OK");

            client.println("Content-type:text/html");

                client.println();

                // the content of the HTTP response follows the header:

                client.print("Romuas Client ");

                // The HTTP response ends with another blank line:

                client.println();

                // break out of the while loop:

                break;

              } else {    // if you got a newline, then clear currentLine:

                currentLine = "";

              }

           } else if (c != '\r') {  // if you got anything else but a carriage return character,

              currentLine += c;      // add it to the end of the currentLine


           if (currentLine.endsWith("GET /green")) {
                  lightGreen();
           }

           if (currentLine.endsWith("GET /red")) {
              lightRed();
            }
          if (currentLine.endsWith("GET /off")) {
              lightOff();
           }

         }
         }

     }

     // close the connection:
     delay(1);
     Serial.println("client disconnected");
}
}
```

Co-funded by the
Erasmus+ Programme
of the European Union

9. It is necessary to also modify the python code (app.py source file) implemented in Project 2: Capture a measure on a button press in order to:

10. Check if a measure is in a valid range

11. Notify to the Arduino board the validity of the measure

12. We need to add new library for the connection with the Arduino web server

```
7    import requests
```

13. Add the declaration of variables needed for managing the threshold and notification to the Arduino board

```
10   enable_external_feedback = True
11   threshold_value= 20
12   threshold_tollerance = 2.0
13   valid_measure_url = "http://192.168.1.80/green"
14   not_valid_measure_url ="http://192.168.1.80/red"
```

- Line 10 enables the notification of the measure validity
- Line 11 sets the initial threshold value
- Line 12 sets the initial threshold tolerance (percentage)
- Lines 13 and 14 set the urls for notify the validity or not validity of the measure

14. Define a function for configuring the threshold and the tolerance

```
@app.route('/config-threshold',methods=['GET', 'POST'])
def config_threshold():
    global threshold_value
    global threshold_tollerance

    if request.method == 'POST':
        threshold_value = int(request.form.get('value'))
        threshold_tollerance = float(request.form.get('tollerance'))

    return render_template('config-threshold.html',
                           t_value=threshold_value,
                           t_tollerance = threshold_tollerance)
```

This function is the request is with method POST update the threshold value and tolerance and return the config-threshold.html template

Co-funded by the Erasmus+ Programme of the European Union

15. Define the template for modifying the threshold value and tolerance

```
<form method="post">
    <p>
        <label>Threshold</label>
        <input type="number"
               name="value"
               required value="{{t_value}}"
               min="10" max="200"/>
    </p>

    <p>
        <label>Tollerance</label>
        <input type="number"
               name="tollerance"
               required
               value="{{t_tollerance}}"
               min="0" max="100" step="0.1"/>%
    </p>
    <button type="submit">Update Threshold</button>
</form>
```

16. Define the function for notify the validity of the measure

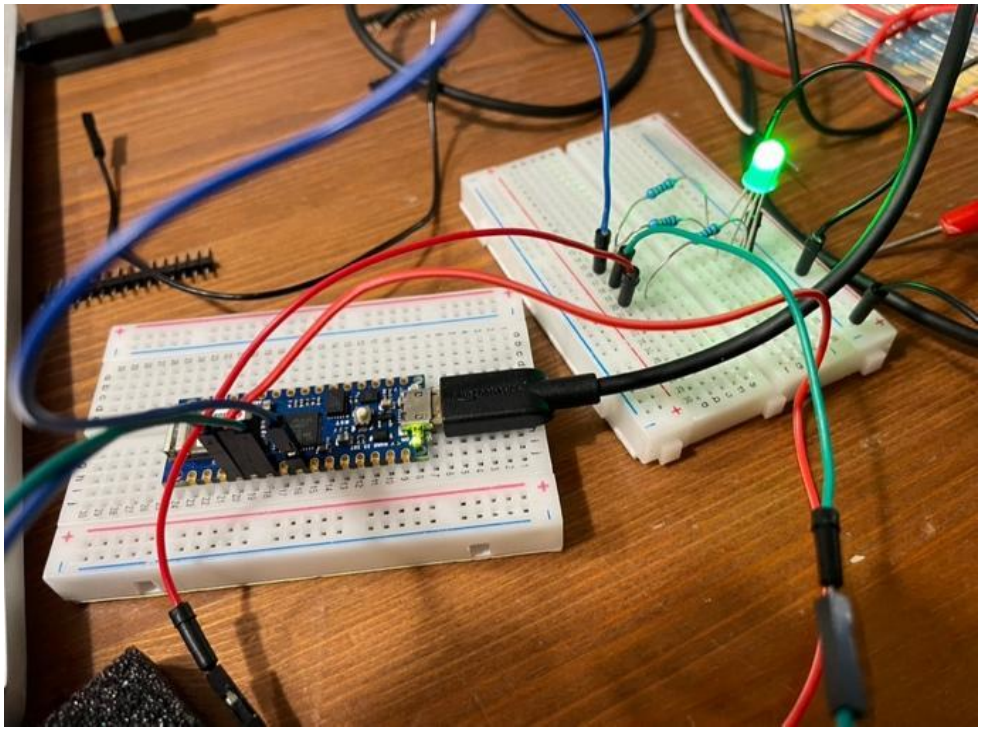
```
#send the feedback to the client (ARDUINO NANO)
def feedback_measure(measure):
    error = threshold_value * threshold_tollerance/100
    feedback_url, valid = valid_measure_url, True if threshold_value - error <= measure <= threshold_value + error else not_valid_measure_url, False
    r = requests.get(feedback_url)

    if r.status_code == 200:

        print('Feedback sent ',valid)
    else:
        print('Error on sending feedback')

```

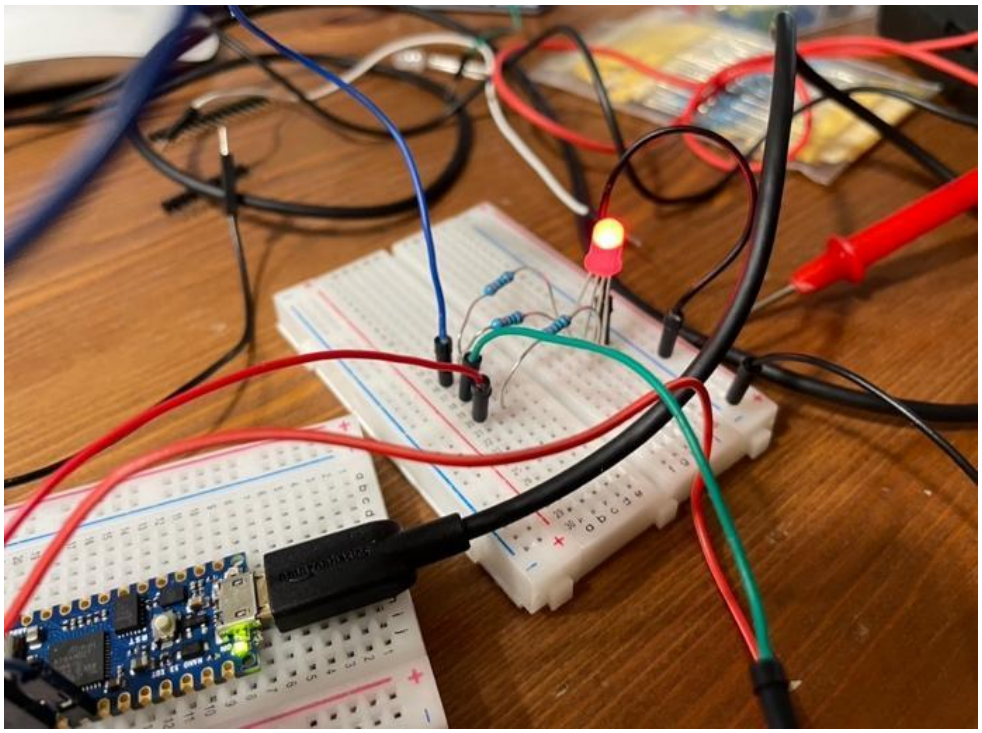
17. Modify the upload Image function for managing the threshold

```
        ocr = romuas_ocr(filePath)
        text = ocr.show_ocr((560,215,230,120))
        samples.append((datetime.now().strftime("%Y %B %d %H:%M:%S"),text,filePath,img_number))

        if enable_external_feedback:
            feedback_measure(int(text))

        return "OK"
```

Co-funded by the
Erasmus+ Programme
of the European Union

18. Working example





Co-funded by the
Erasmus+ Programme
of the European Union

Project 4 Send measures to cloud

DIFFICULTY	Hardware Easy Medium Hard  Software Easy **Medium** Hard 
SOFTWARE REQUIREMENTS	Arduino IDE installed on Windows, Linux or macOS, Tesseract Engine, **Arduino Cloud**
HARDWARE REQUIREMENTS	Project 3 requirements
TIME	about 20 minutes if the projects 3 has already been completed

This project is an extension of Project 3: Switch on a green led on a *correct* measure with the addition of the integration with the Arduino Cloud for storing the recognized measure and to have a real-time overview of the Universal Measure Reader status.

In Figure 24 is shown a high-level overview of the project: basically, it is the same architecture of project 3, but when the validity feedback is sent to the Arduino Nano board, the same notification is sent to the cloud.

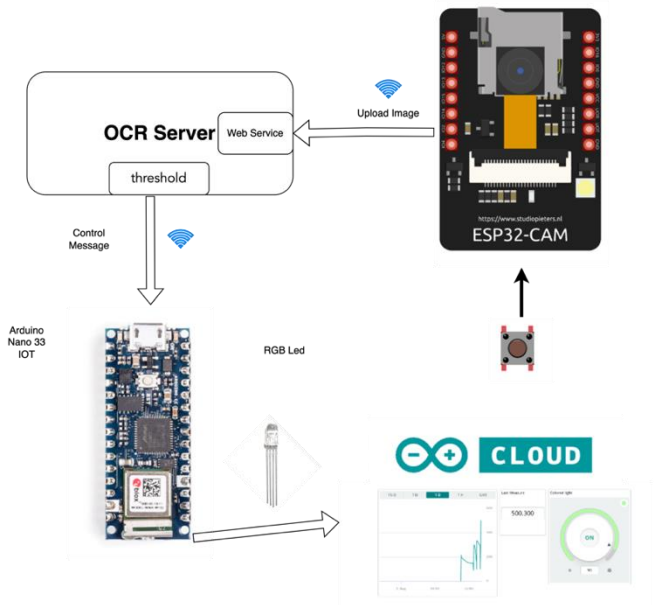


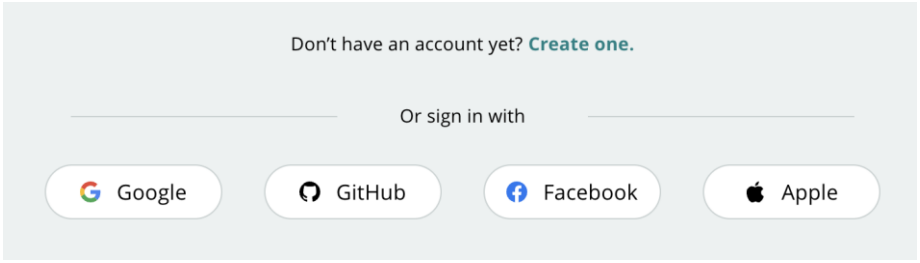
Figure 24 Project 4 high-level overview

Co-funded by the Erasmus+ Programme of the European Union

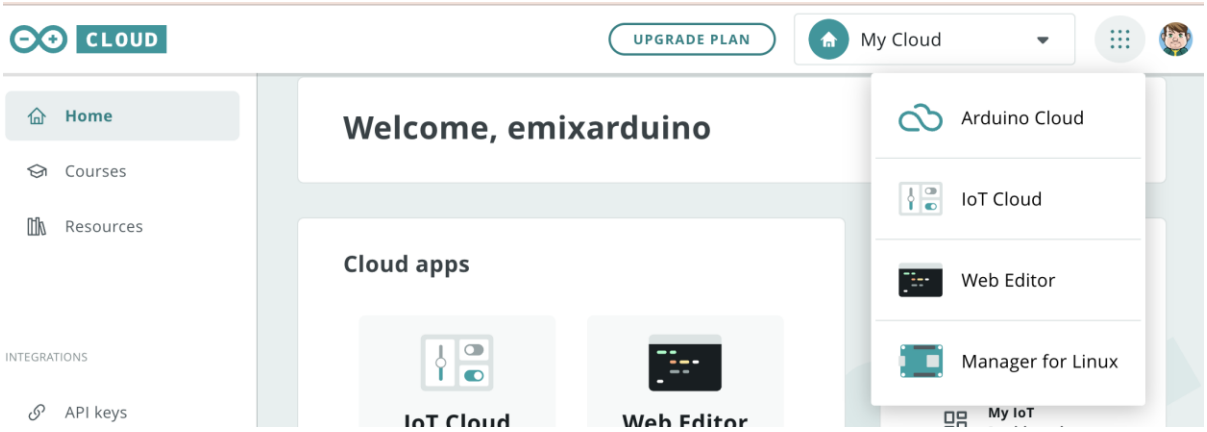
1. First of all, you need to complete the wiring proposed for Project 3 (see Figure 22 and Figure 23)

2. Sign in or register to Arduino Cloud
   - Navigate to https://cloud.arduino.cc/
   - Click on Sign In

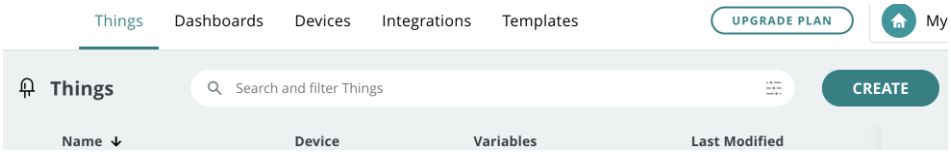
   - Click on Create one, or make used the sigle signon with other provider



3. Once you are registered an signed in you can access to the Arduino Cloud home and select IoT Cloud functionality from the top right menu

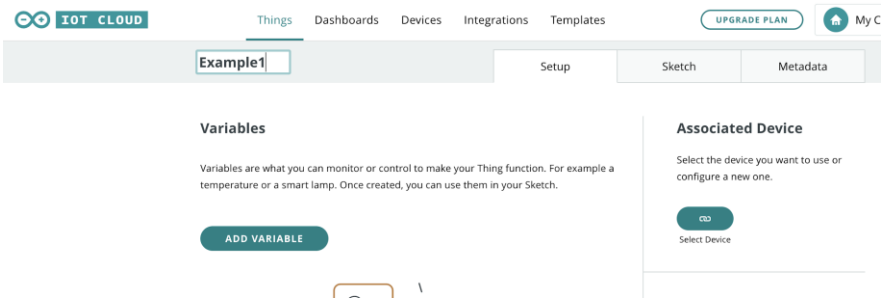


4. We need to create a new virtual device that will map our real Arduino board.
   - Click on Things menu
   - Click on Create button

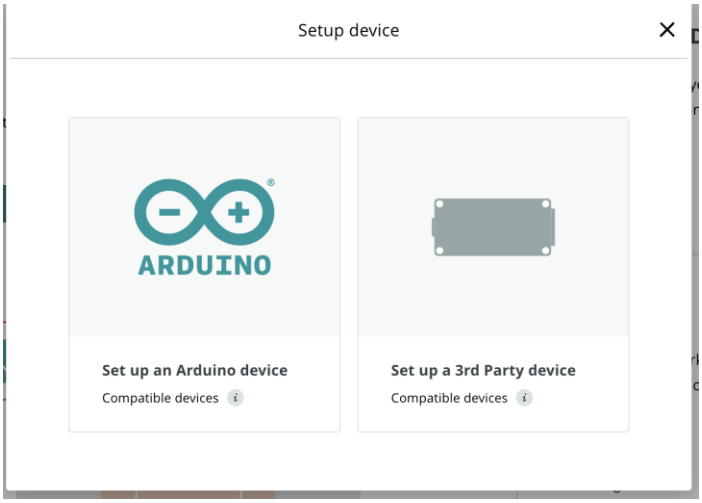


Co-funded by the Erasmus+ Programme of the European Union

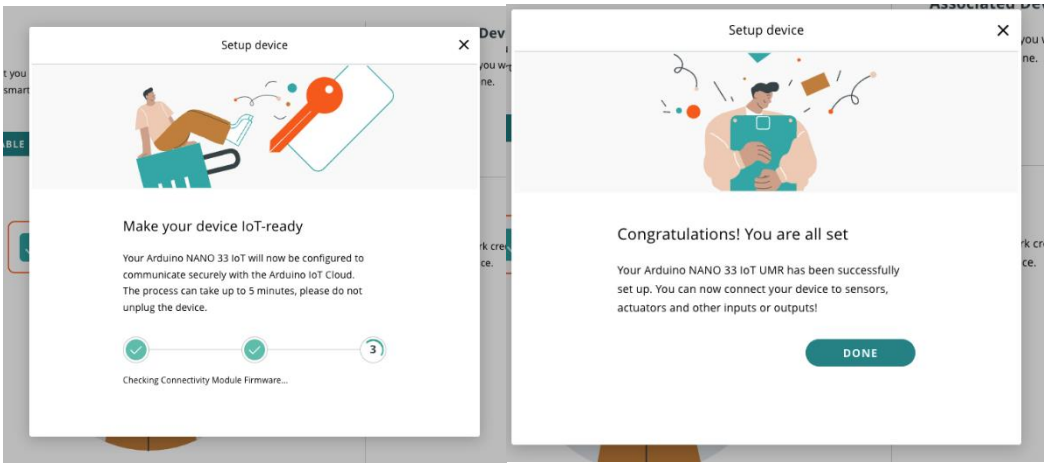
- Select a name for the device



- Click on Select Device on the right sidebar. A wizard for selecting the device will start



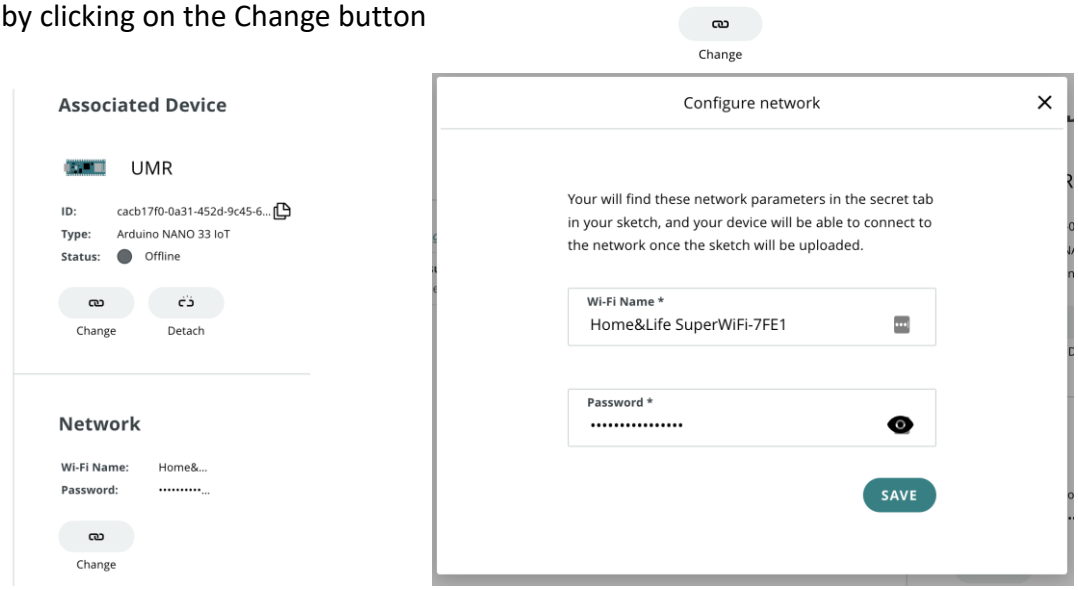
- You need to connect the Arduino Board to your computer, and you need have already installed an Arduino Agent [11]
- The connection dialog performs some operation and generate a sketch file



[11] Arduino Agent is a middleware software that allows access and programming Arduino boards via browser editor. More information about how to install the Arduino Create Agent is provided at https://support.arduino.cc/hc/en-us/articles/360014869820-Install-the-Arduino-Create-Agent

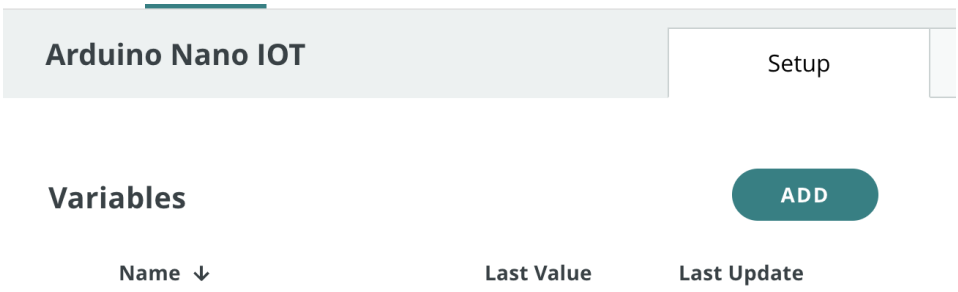
Co-funded by the Erasmus+ Programme of the European Union

5. In the Setup tab of the device page, it is possible to define your Wi-Fi connection parameters by clicking on the Change button

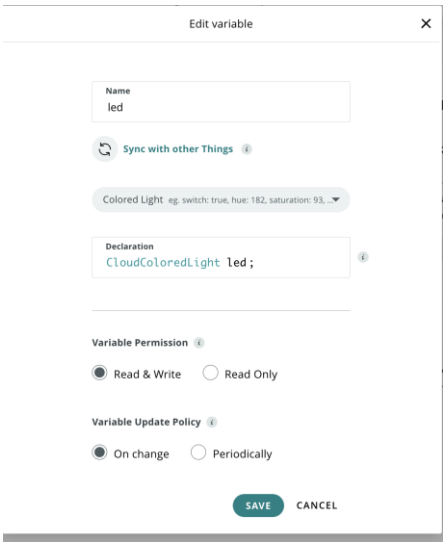


6. We need to create two variables for storing the current read measure and its validity status (led color).

- In the Setup section of our device just click on Add button

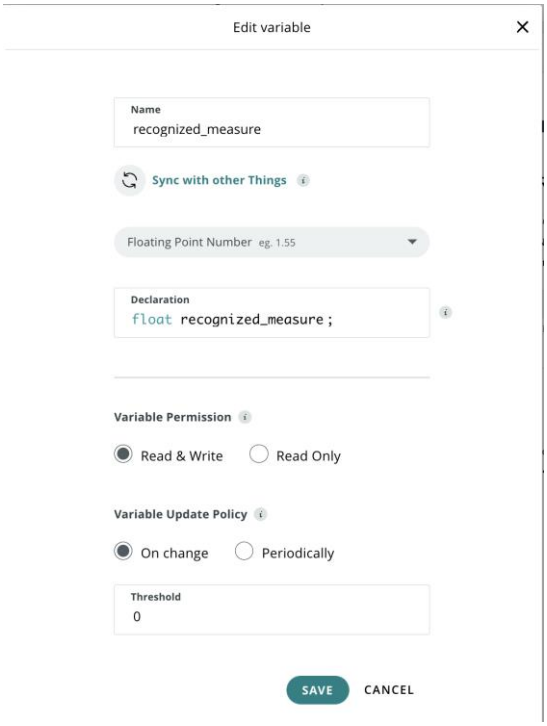


- For the Led variable choose the following parameters



Co-funded by the Erasmus+ Programme of the European Union

- For the recognized measure insert the following data



- The system automatically generate a Sketch

7. We need to modify the generated Sketch by merging the code written for the previous project.
   - Since the code is quite similar to the previous project, we are going to comment only the differences
   - Access the Sketch tab and click on Open full editor

   - 4 files will be shown

   - We need to modify the first one

Co-funded by the Erasmus+ Programme of the European Union

8. If you look the file thinProperties.h you can notice that the two variables we create in step 6 are included. Changing the values of such object will automatically notify it to the Arduino cloud.

```
float recognized_measure;
CloudColoredLight led;

void initProperties(){

  ArduinoCloud.addProperty(recognized_measure, READWRITE, ON_CHANGE, onRecognizedMeasureChange);
  ArduinoCloud.addProperty(led, READWRITE, ON_CHANGE, onLedChange);

}
```

9. Come back to the .ino file, we need to import the required libraries and create the variables

```
#include <WiFiNINA.h>
#include "thingProperties.h"


const int RED =  2;   //pin D2
const int GREEN = 3;   //pin D3
const int BLUE =5;    //pin D5


WiFiServer server(80);
```

10. Define a function to switch of the RGB led

```
void lightOff(bool notify=true){
  analogWrite(RED, LOW);
  analogWrite(GREEN, LOW);
  analogWrite(BLUE, LOW);
  Serial.println("OFF");
  if(notify)
    led=ColoredLight(false,100,0,100);
}
```

This function has an optional parameter set true by default that indicates if it is necessary to change the variable mapped with cloud virtual device.

11. Define a function for change the RGB led color

```
void setColor(int r,int g, int b){
  analogWrite(RED,r);
  analogWrite(GREEN,g);
  analogWrite(BLUE,b);

}
```

Co-funded by the Erasmus+ Programme of the European Union

12. Define the functions for change the color to Red and to Green

```
void lightRed(bool notify=true){
  analogWrite(RED, HIGH);
  analogWrite(GREEN, LOW);
  analogWrite(BLUE, LOW);
  Serial.println("light Red");
  if(notify)
    led=ColoredLight(true,0,100,100);

}

void lightGreen(bool notify=true){
  analogWrite(RED, LOW);
  analogWrite(GREEN, HIGH);
  analogWrite(BLUE, LOW);
  Serial.println("light Green");
  if(notify)
    led=ColoredLight(true,120,100,100);
}
```

13. Define the setup function that will be executed once at the device start (or reset)

```
62 void setup() {
63   // Initialize serial and wait for port to open:
64   Serial.begin(9600);
65   // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
66   delay(1500);
67
68   // Defined in thingProperties.h
69   initProperties();
70
71   // Connect to Arduino IoT Cloud
72   ArduinoCloud.begin(ArduinoIoTPreferredConnection);
73
74   /*
75     The following function allows you to obtain more information
76     related to the state of network and IoT Cloud connection and errors
77     the higher number the more granular information you'll get.
78     The default is 0 (only errors).
79     Maximum is 4
80   */
81   setDebugMessageLevel(2);
82   ArduinoCloud.printDebugInfo();
83   // attempt to connect to WiFi network:
84
85   server.begin();
86 }
```

Co-funded by the
Erasmus+ Programme
of the European Union

14. Define a function for managing HTTP request (sent by the ESP32-CAM board)

```
void HandleClientRequest(){
  // listen for incoming clients
  WiFiClient client = server.available();   // listen for incoming clients

  if (client) {                             // if you get a client,
    Serial.println("new client");           // print a message out the serial port
    String currentLine = "";                // make a String to hold incoming data from the client

    while (client.connected()) {            // loop while the client's connected

      if (client.available()) {             // if there's bytes to read from the client,

        char c = client.read();             // read a byte, then
        if (c == '\n') {                    // if the byte is a newline character

          // if the current line is blank, you got two newline characters in a row.
          // that's the end of the client HTTP request, so send a response:

          if (currentLine.length() == 0) {

            // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)
            // and a content-type so the client knows what's coming, then a blank line:

            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println();
            // the content of the HTTP response follows the header:
            client.print("Romuas Client ");
            // The HTTP response ends with another blank line:
            client.println();
            // break out of the while loop:
            break;

          } else {    // if you got a newline, then analyze the line and clear currentLine:

            if (currentLine.indexOf("GET /green")>=0) {
              lightGreen();
            }

            if (currentLine.indexOf("GET /red")>=0) {
              lightRed();
            }
            if (currentLine.indexOf("GET /off")>=0) {
              lightOff();
            }
            if (currentLine.indexOf("GET /?value=")>=0) {
              int rightIndex = currentLine.lastIndexOf(" ");
              String measureText = currentLine.substring(12,rightIndex);
              Serial.println("Measure="+measureText);
              recognized_measure = measureText.toFloat();
            }
            Serial.println(currentLine);


            currentLine = "";

          }

        } else if (c != '\r') {  // if you got anything else but a carriage return character,

          currentLine += c;      // add it to the end of the currentLine
        }
        }
      }
```

Co-funded by the Erasmus+ Programme of the European Union

```
152
153       // close the connection:
154       delay(1);
155       Serial.println("client disconnected");
156   }
157
158
159   }
```

As difference in comparison with Project 3, this code manages also an kind of HTTP request: GET /?value.

This request allows to send to the Arduino board the value of the recognized measure. The update of the recognized_measure variable (line 137) consents the automatic notification of the data to the Arduino cloud.

15. Define the main loop

```
162 void loop() {
163   ArduinoCloud.update();
164   // Your code here
165   HandleClientRequest();
166
167 }
```

16. Define the onLedChange function that will be triggerd when the led color value is changed on the cloud web page.

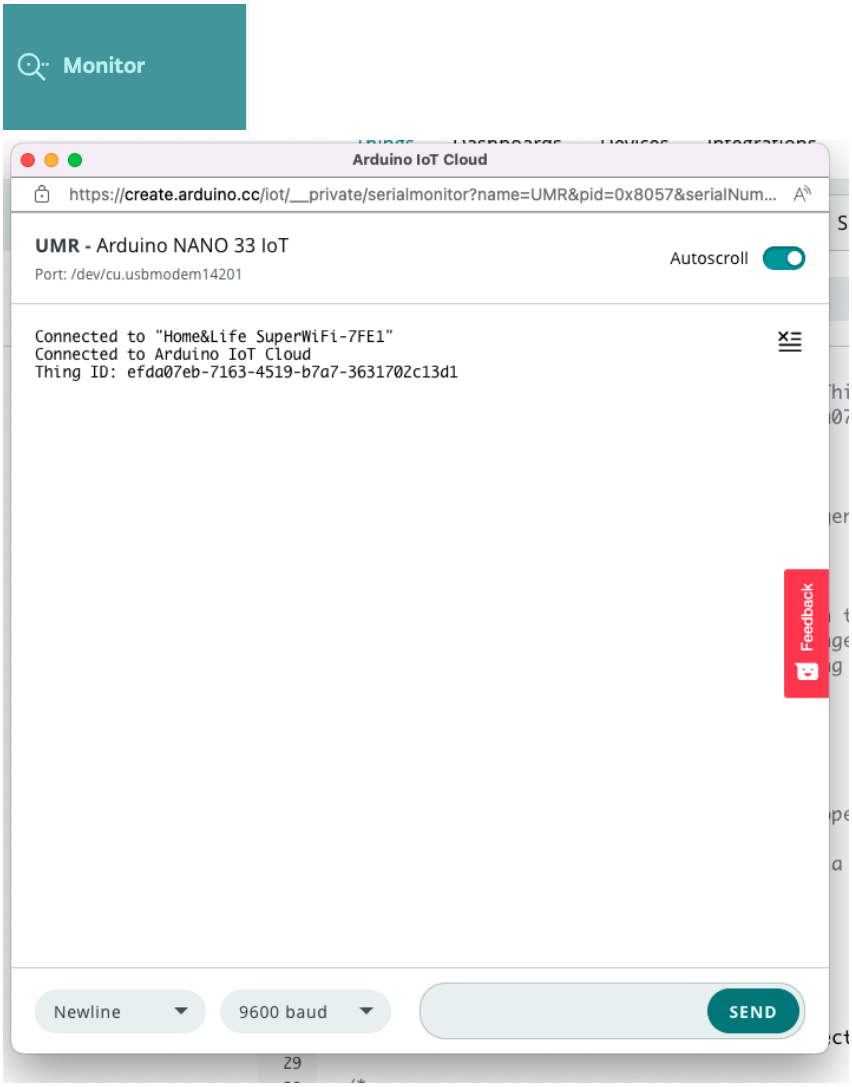
```
200 void onLedChange()  {
201   auto colorValue = led.getCloudValue();
202   if(colorValue.swi){
203     uint8_t r,g,b;
204     colorValue.getRGB(r,g,b);
205     setColor(r,g,b);
206
207   }else{
208     lightOff(false);
209
210   }
211 }
```

Sincerely, changing the led color vi cloud web interface is not strictly necessary for the project purpose, but it is reported hero for showing an example of two way integration (Device to Cloud and Cloud to Device).

17. Upload the sketch to the device
    - It is necessary to press the arrow button 

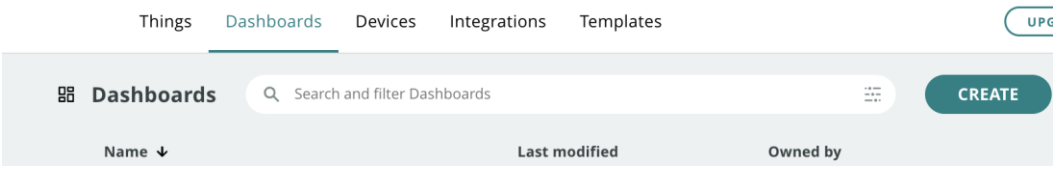
Co-funded by the Erasmus+ Programme of the European Union

18. After uploading the sketch you can open the monitor functionality

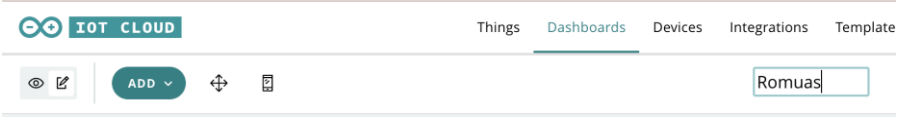


19. Come back to IoT Cloud and create a new Dashboard

- Navigate to the Dashboards menu and click the Create button



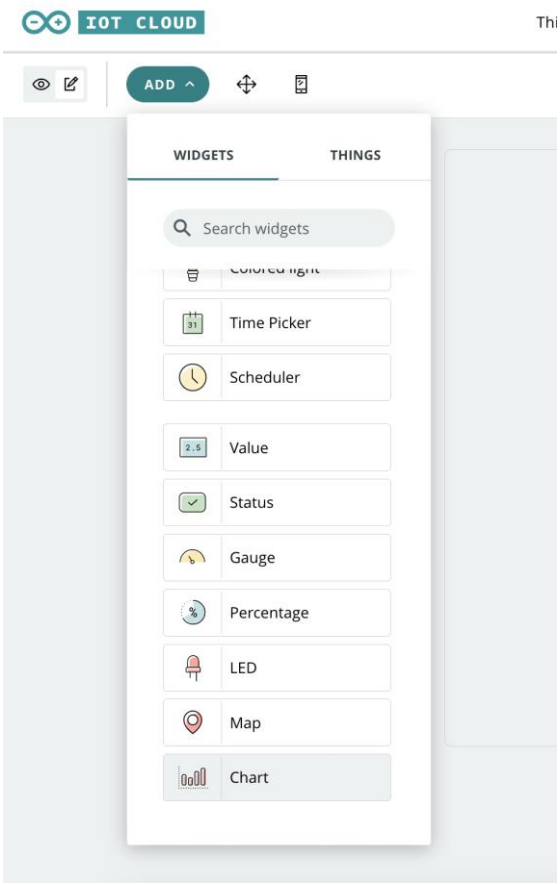
- Select a name for the Dashboard (e.g. Romuas)



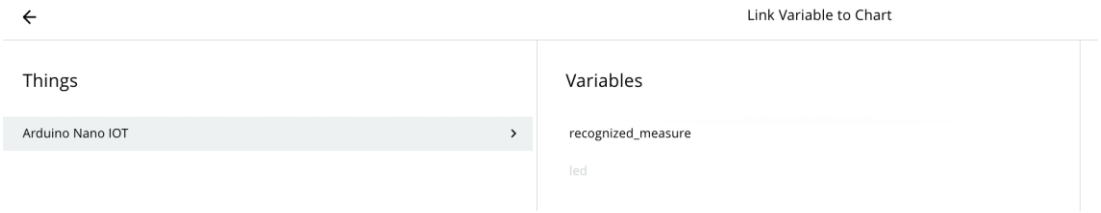
20. Create a chart widget

Co-funded by the Erasmus+ Programme of the European Union

- Click the Add button



- Link the chart to recognized_measure variable

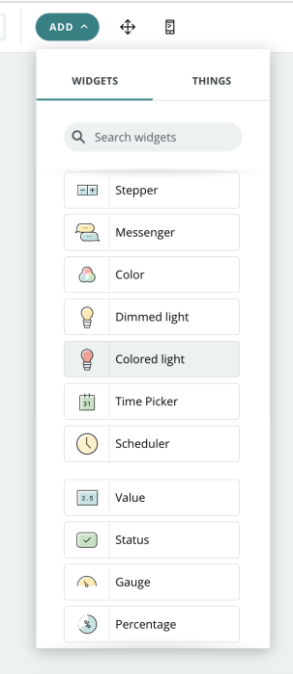


- The chart will report the data notified by Arduino board to the Cloud

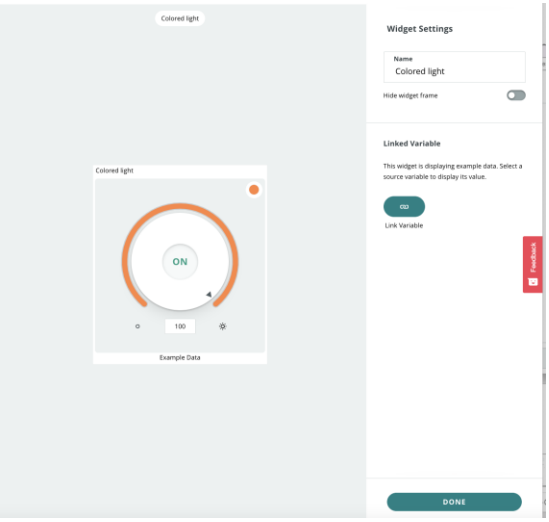
Co-funded by the
Erasmus+ Programme
of the European Union

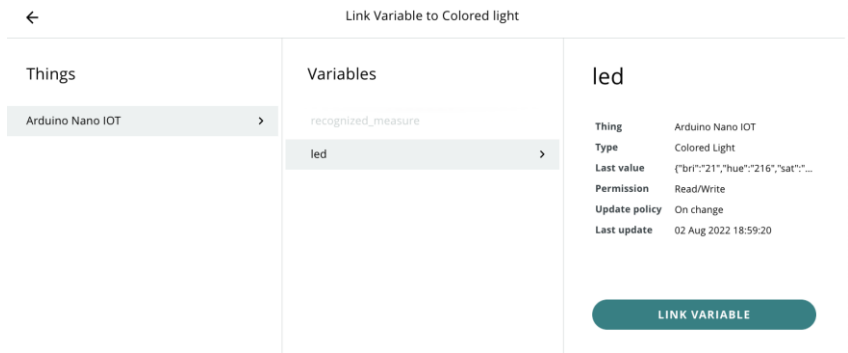
21. Add a Colored light widget

- Click on Add menu and select colored light



- Link the led to the “led” variable

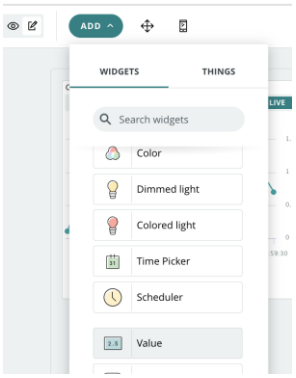




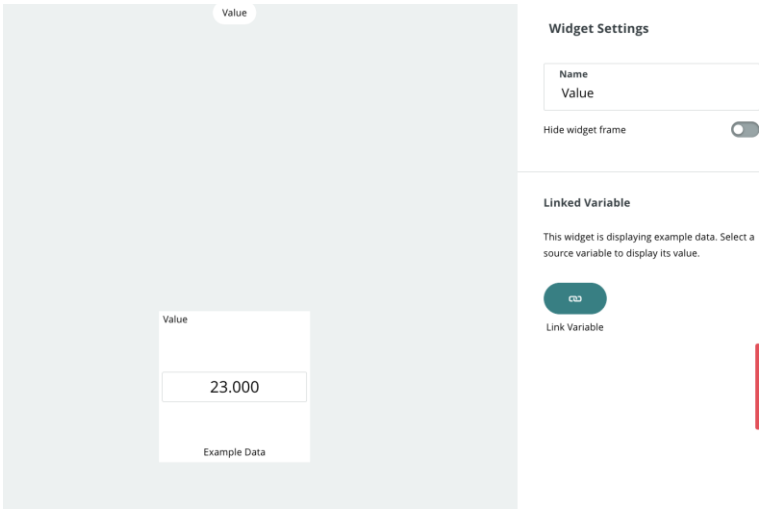
Co-funded by the Erasmus+ Programme of the European Union

22. Add a value widget

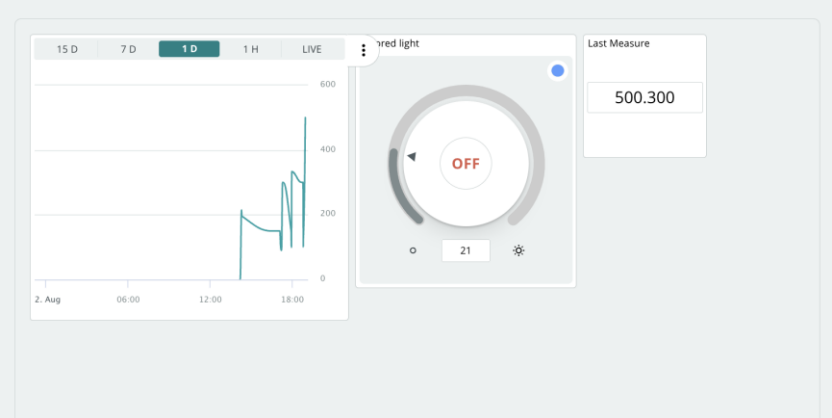
- Click on Add menu and select “Value”



- Link it to the recognized_measure variable



23. The final layout of the dashboard should be the following



The data are update real-time and the connection between Arduino board and the cloud is two ways, if you change the color of the led in the dashboard the color will be changed on the real device.

Co-funded by the Erasmus+ Programme of the European Union

24. We need to modify the ocr-server for sending the recognized meausure value to the Arduino Nano board.

- Modify the app.py file by adding the measure_url variable

```
15    measure_url ="http://192.168.1.80/?value="
```

- Modify the feedback_measure function

```
127  #send the feedback to the client (ARDUINO NANO)
128  def feedback_measure(measure):
129      error = threshold_value * threshold_tollerance/100
130      feedback_url, valid = valid_measure_url, True if threshold_value - error <= measure <= threshold_value + error \
131          else not_valid_measure_url, False
132      r = requests.get(feedback_url)
133
134      if r.status_code == 200:
135
136          print('Feedback sent ',valid)
137      else:
138          print('Error on sending feedback')
139
140      r = requests.get(measure_url+measure)
141
142      if r.status_code == 200:
143
144          print('Feedback sent ',valid)
145      else:
146          print('Error on sending feedback')
```

- Lines from 140 to 146 send the measure the Arduino Nano 33 IoT board that will notify it to the Cloud

Co-funded by the Erasmus+ Programme of the European Union

EXAMPLES OF LABORATORY EXPERIENCES TO CONDUCT BEFORE AND AFTER THE UPDATE

Some experiments have been conducted to understand the effect of the implementation of some common instruments used in a chemistry lab, as a lab balance ad and heater.
Data are collected manually and analyzed with a calculator and some calc programmes.
The explanation of these experiment is outlined below.

UPDATE OF A LAB BALANCE AND A HEATER

An aged balance is used to conduct two experiments, one is focused to evaluate the density of some materials, the second one to determinate the quantity of organic part of some soils. The second experiment is conduct using both lab balance and heater. After the collection, the data are analysed. A third experiment is conduct only with the heater, after the update.

1. Evaluation of the density of different materials

To assess their apparent density, six different materials were weighed. Initially, a water sample was used to check the accuracy of the balance and the volume that the beaker can contain. A known volume of different material was weighted on the lab balance. The sample were tap water (H_2O-test), sodium chloride (NaCl), saccharose ($C_{12}H_{22}O_{11}$), lithium carbonate ($LiCO_3$), copper powder (Cu), sand.

The container was a glass beaker that may hold a maximum volume of 50 cm^3, graduated on 10 cm^3, 25 cm^3, 40 cm^3 and 50 cm^3.

Firstly, 40 ml of water was weighted on the lab balance to verify the accuracy on the lab balance (hundredth of a gram): the weight was perfectly 40 g. Indeed, the density of the water is 1g/cm^3 and the bulk density is not to considered, because the water is in liquid state.

Co-funded by the Erasmus+ Programme of the European Union

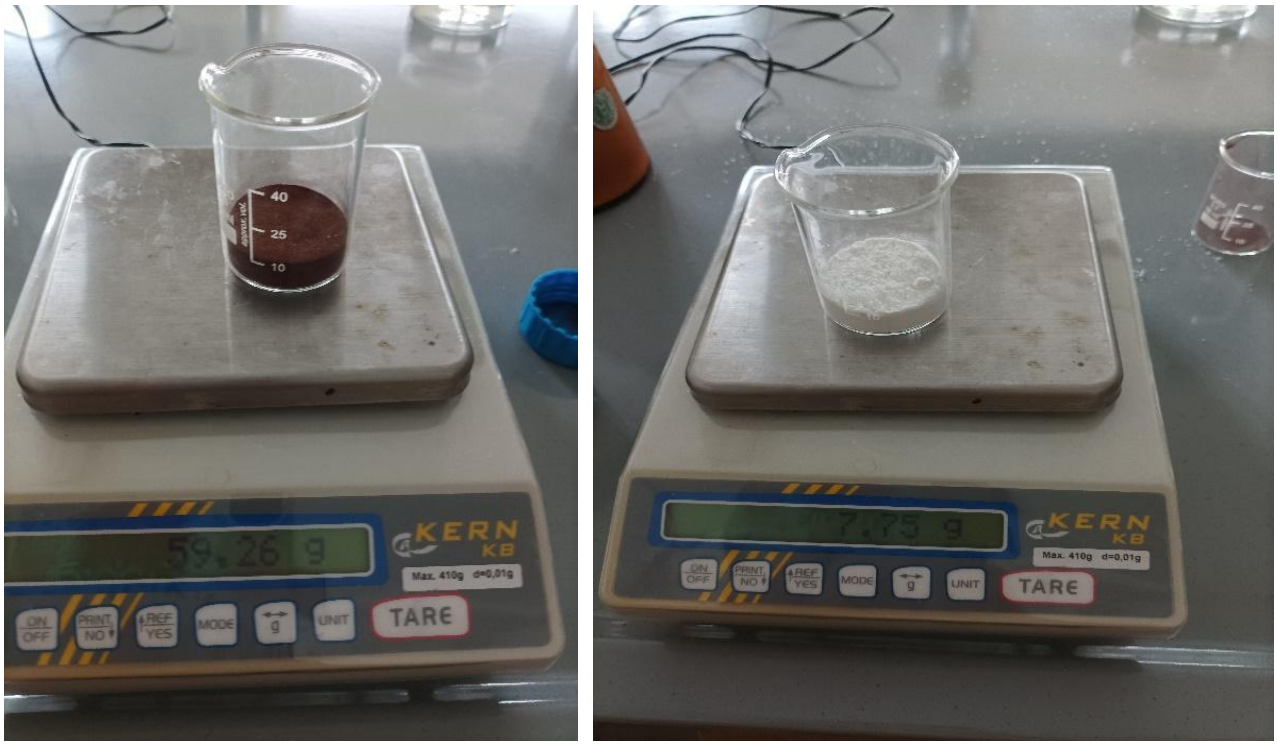


Fig. 1-2: copper powder and lithium carbonate samples

In a second time, the materials were weighted and the balance gave back the following data.

MATERIAL	VOLUME	WEIGHT
tap water	40 ml	39,96 g
sodium chloride	10 ml	11,91 g
saccharose	10 ml	9,40 g
lithium carbonate	10 ml	7,75 g
copper powder	10 ml	59,26 g
sand	10 ml	19,41 g

After data collection, data were analysed with computer software (as Microsoft Excel), manually entering the values. Density is defined as mass divided by volume, or mass per unit volume. They have been considered experimentally evaluated density values to obtain the porosity of solid materials.

Density is calculated with the following formula: $\sigma = M/V$, where σ is density, M is mass, and V is volume. Density is commonly expressed in units of grams per cubic centimetre. For example, the density of water is almost 1 gram per cubic centimetre (g/ cm^3) and continental crust is 2,7 grams per cubic centimetre (g/ cm^3).

Porosity is the amount of empty space inside of a solid dry material and it depends of shape of grains, which determines shape and size of pores, and the size of the grains. Concerning the

Co-funded by the Erasmus+ Programme of the European Union

evaluation of porosity it has to consider the real density and bulk density (measured with the balance). The porosity is calculated with the following formula:

$$P= (1-Pb/Pd)*100$$

Where p is porosity of the sample, Pb is bulk density and Pd is the particle density (experimental value).
After this, porosity values are calculated, estimated with experimental particle density data's (see attachments).

MATERIAL	**WEIGHT**	**VOLUME**	**BULK DENSITY**	**PARTICLE DENSITY**	**POROSITY %**
	g	*ml*	*σ = M/V (g/cm^)g/cm^3*		*P= (1-Pb/Pd)*100*
tap water	39,96	40	1,00	1	0,1
sodium chloride	11,91	10	1,19	2,16	44,86
saccharose	9,40	10	0,94	1,59	40,86
lithium carbonate	7,75	10	0,78	2,1	63,10
copper powder	59,26	10	5,93	8,9	33,42
sand	19,41	10	1,94	2,7	28,11

Co-funded by the Erasmus+ Programme of the European Union

2. Experiment repeated after the update

After the update, data are automatically analysed again with Arduino. The operator has only to set the lab balance and put materials on that.

Following the description, step by step, of the process.

Firstly, the operator sets Arduino to collect data (positioning the camera, that acquires images). The samples are weighted on the lab balance, the operator the button on the left and the associated programme acquires data, that are visible on Excel sheet.

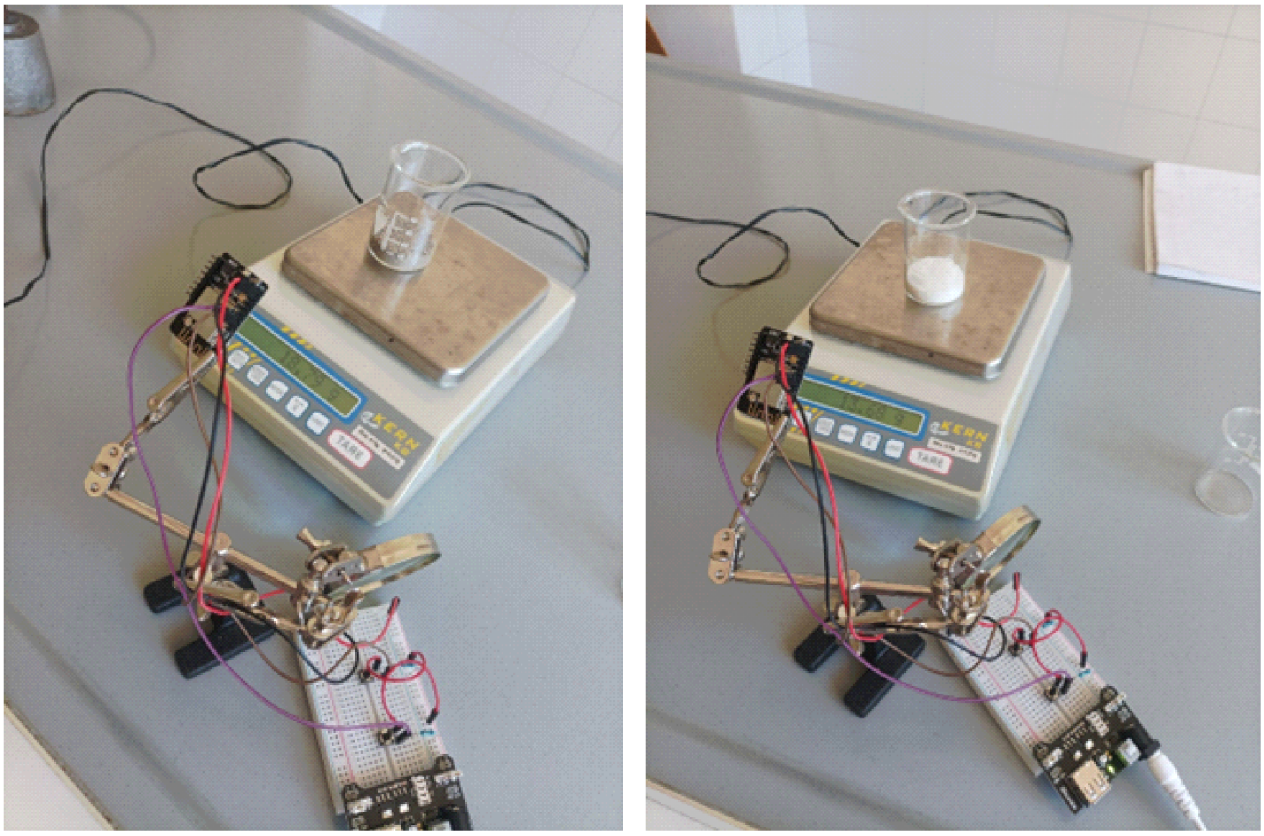


Fig. 3-4: sand and sodium chloride samples on the balance, after the update

Elaboration of the data:

EXPERIMENT	WEIGHT (g)	SUBSTANCE	
Measure 1	48,03	Tap Water	1st measure
Measure 4	18,77	Sand	1st measure
Measure 7	13,63	Sodium Chloride	1st measure
Measure 10	10,24	Saccharose	1st measure

Co-funded by the Erasmus+ Programme of the European Union

Measure 13	7,95	Lithium Carbonate	1st measure
Mesure 16	59,99	Copper powder	1st measure

Later, data are analysed and compared to the values obtained with the previous experiment.

MATERIAL	WEIGHT	VOLUME	BULK DENSITY	PARTICLE DENSITY	POROSITY %
	g	*ml*	*σ = M/V (g/cm^)g/cm^3*		*P= (1-Pb/Pd)*100*
tap water	48,04	50	0,96	1	--
sodium chloride	13,63	10	1,36	2,16	36,90
saccharose	10,24	10	1,02	1,59	35,60
lithium carbonate	7,95	10	0,80	2,1	62,14
copper powder	59,99	10	6,00	8,9	32,60
sand	18,77	10	1,88	2,7	30,48

The values are slightly different because of their porosity, in fact it depends of the dimension of grains.

To conclude, the experiment conduct with Arduino is quicker because the operator, after settings it, has only to prepare the samples and weight. The collection of the data is automatic and the elaboration is made in a second time.
That device is useful in case of a large number of consequential measurements, because the operator mustn't interrupt his work to take note of the data and it avoids any distraction mistakes.

Co-funded by the Erasmus+ Programme of the European Union

Bibliography

1. **Santos, Sara.** ESP32 Camera Dev Boards Review and Comparison. *Maker Advisor.* [Online] 04 28, 2020. https://makeradvisor.com/esp32-camera-cam-boards-review-comparison/.

2. **Santos, Rui.** $7 ESP32-CAM with OV2640 Camera. *Maker Advisor .* [Online] 9 6, 2021. [Cited: 06 23, 2022.] https://makeradvisor.com/esp32-cam-ov2640-camera/.

3. **Santos, Sara.** ESP-EYE: ESP32-based board for AI. *Maker Advisor.* [Online] 04 25, 2021. [Cited: 06 23, 2022.] https://makeradvisor.com/esp-eye-new-esp32-based-board/.

4. **Santos, Sara.** TTGO T-Journal ESP32 Camera Development Board Review. *Maker Advisor.* [Online] 04 26, 2021. [Cited: 06 23, 2022.] https://makeradvisor.com/ttgo-t-journal-esp32-camera-board-review/.

5. **Ayibiowu, Ayo.** TTGO T-CAMERA IS AN ESP32 CAM BOARD WITH OLED AND AI CAPABILITIES. *Electronics-Lab.* [Online] 01 30, 2019. [Cited: 06 23, 2022.] https://www.electronics-lab.com/ttgo-t-camera-esp32-cam-board-oled-ai-capabilities/.

6. Upload Code to ESP32-CAM AI-Thinker using ESP32-CAM-MB USB Programmer. *Random Nerd Tutorial.* [Online] [Cited: 06 24, 2022.] https://randomnerdtutorials.com/upload-code-esp32-cam-mb-usb/.

7. Arduino Ide Dowloads. *Arduino.cc.* [Online] [Cited: 06 24, 2022.] https://www.arduino.cc/en/software.

8. Editing the ESP32-CAM Camera Web Server HTML. *Robot Zero One.* [Online] 04 25, 2019. [Cited: 06 25, 2022.] https://robotzero.one/esp32-cam-custom-html/.

9. The technology for converting books and documents into electronic files.. *Software Sources.* [Online] [Cited: July 7, 2022.] https://www.abbyy.co.il/?categoryId=72050&itemId=168963.

10. Linvill, John G. *Optacon Final Report.* s.l.: Stanford Electronics Iaboratories , 1973. Stanford, California 943O5.

11. Installation. *Tesseract OCR.* [Online] 2022. https://tesseract-ocr.github.io/tessdoc/Installation.html.

12. IEC. *Degrees of protection provided by enclosures (IP Code).* 2013. ISBN 978-2-8322-1086-4.

13. Arduino case 3d print. *All3DP.* [Online] https://all3dp.com/2/arduino-case-3d-print-3d-models/.

14. Santos, Raul. ESP32-CAM Post Images to Local or Cloud Server using PHP (Photo Manager). *Random Nerd Tutorials.* [Online] https://randomnerdtutorials.com/esp32-cam-post-image-photo-server/.

Co-funded by the Erasmus+ Programme of the European Union

Co-funded by the
Erasmus+ Programme
of the European Union

ROMUAS
KILLS
RDUINO
SING
MACHINES
OUTDATED
ECOVER

Social innovation

C3

Expert program toolkit

PR1

3) Social innovation

I. Theoretical part

1. Introduction
2. Brief historical development of technology
3. Social skills
   - 3.1. Assertiveness
   - 3.2. Communication skills
   - 3.3. Development of social skills
4. Impact of technology on social skills
5. Interactive methods for the development of social skills
6. Conclusion

Bibliography

II. Interactive methods to develop innovative skills

1. Discover my element
2. From inspiration to action
3. Bibliography

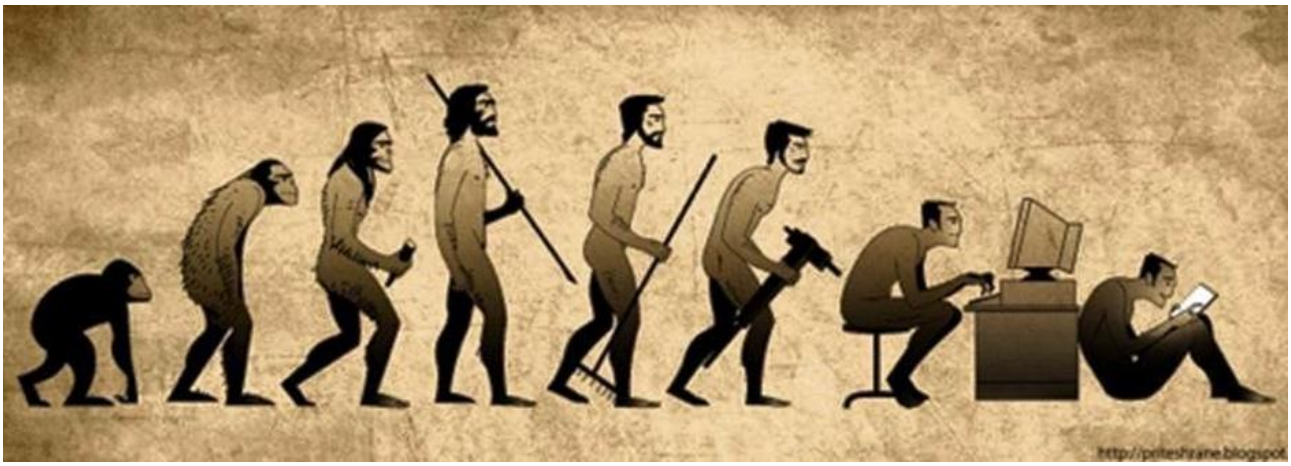
Co-funded by the
Erasmus+ Programme
of the European Union

I. Theoretical part

"The saddest aspect of life right now is that science gathers knowledge faster than society gathers wisdom." Isaac Asimov

1. Introduction

2.5 million years ago, people started making stone tools to conquer nature and survive in harsh conditions. This is when the development of technology begins. From the beginning, man developed technology to adapt nature to his needs and survival. Over time, the development of technology accelerated because man tried to simplify, facilitate, and speed up numerous processes. From the middle of the 18th century, when the first industrial revolution began, the development of technology accelerated, but also spread to numerous areas of human activity. Today, technology is no longer viewed only as a "tool" that is in a subordinate relationship with man. Man develops and uses technology, and technology affects the development and actions of man. In this relationship, both man and technology benefit, and nowadays it is seen as a kind of symbiosis.



Source: Izvor:https://www.facebook.com/TCHADDN/photos/a.464912620519513/464912600519515/?type=3

It is quite clear how man influences technology because we ourselves witness the continuous acceleration of progress in various fields, from agriculture and industry, all the way to medicine, communications, and the space industry. However, a question that has not been asked since the very beginning of the development of technology is how does technology affect the psychological development and psychological processes of man? What are the areas of human development and the processes that technology affects? Is its influence exclusively positive or does it also have negative effects?

Co-funded by the Erasmus+ Programme of the European Union

2. Brief historical development of technology

From the very beginnings of development, the purpose of technology was to facilitate survival, and later to facilitate and speed up production. During the first industrial revolutions and the increasing automation of production, many workers lost their jobs. This caused dissatisfaction among them, but also encouraged them to "retrain". Retraining included the acquisition of new skills and knowledge that enabled these workers to find new employment and ensure survival for them and their families. A direct consequence of the development of technology is the increasing complexity of the requirements of certain occupations. At the same time, the areas of technology development began to intertwine more and more, and the development in one area greatly influenced the development in another area. The development of technology began to be increasingly influenced by historical and geopolitical relations. The Second World War influenced the development of the first electronic computer because it was necessary to perform numerous computational operations when manufacturing many war tools. With the end of II. World War, the Cold War begins, which accelerated the development of technology, all because of the mutual competition between the West and the East. One of the areas where the greatest progress has been made is the area of communication systems that have enabled the development of the military and space industry. It was the development of communication systems at that time that laid the foundation for the development of mobile technology and the Internet, with the development of which, technology was increasingly embedded in everyday life.

Source: https://fredcavazza.net/2022/06/14/panorama-des-medias-sociaux-2022/#more-38456

The first SMS to a mobile phone was sent in 1992, and it contained a premature Christmas card. At that time, desktop computers were increasingly being used in the business world, and they were slowly entering home use as well. Only 3 years later, the head of Microsoft (then the largest manufacturer of desktop computers), announced that their products would completely turn to the Internet. Since then, the digital revolution has started! The Internet is becoming available to an increasing number of people and is used to search for various information, read and follow news from around the world, listen to music, watch movies, contact friends and relatives, etc. The Internet is becoming a "place" where people begin to spend up to 30% of your free time (according to Antolović, 2013). At the same time, mobile phones are becoming more and more "smart", they have more and more functions that enable them to act as minicomputers. SMS messages are used to send greetings, pay parking tickets, participate in prize games, etc. Services such as Viber,

Co-funded by the Erasmus+ Programme of the European Union

WhatsApp, Messenger, etc. are also being developed, which allow us to send unlimited messages provided we have access to the Internet. However, the real global communication phenomenon is social networks such as Facebook, Instagram, TikTok, etc. Considering that man is a social being, it is not at all surprising that he is attracted to social communication. On social networks, we try to get closer, connect and even make friends with new people, we exchange different information. "The main characteristic of all social networks are universal means of communication and techniques, as well as a common interest that holds groups of people together and thus gives popularity to all social networks today" (Grbavac, 2014). By 2020, social interactions via the Internet, mobile phones, and social networks will mostly involve verbal, written communication. With the declaration of the coronavirus pandemic and the recommendations of medical authorities to reduce social contacts to a minimum, the world begins to function online. Therefore, social interactions are also being transferred to the online environment. At the same time, people's need for social contact is not only satisfied by written communication, but various communication platforms are increasingly used, such as Zoom, MS Teams, Webex, etc. The use of communication platforms enables faster communication with one or more people, with face-to-face contact.

In parallel with the development and increased use of the Internet, smartphones, social networks and communication platforms, more and more research are emerging that deals with the impact of digital technologies on various aspects of human development, especially on social interactions and social skills.

3. Social skills

Social skills are "a set of learned abilities that enable an individual to interact competently and appropriately in a given social context. The most identified social skills in Western cultures include assertiveness, coping, communication and friendship skills, interpersonal problem solving, and the ability to regulate one's cognitions, feelings, and behaviors." (APA Dictionary, 2022).

Source: https://www.shutterstock.com/image-vector/social-skills-we-use-everyday-interact-1933402433

3.1. Assertiveness

Assertiveness is defined as "an adaptive style of communication in which a person directly expresses his emotions and needs, while maintaining respect for others. A lack of assertiveness can contribute to depression and anxiety, while maladaptive approaches to assertiveness can manifest as aggression." (APA Dictionary, 2022).
As can be seen in the picture, two dimensions are crucial for different communication styles: openness of communication and respect for others. Between these two dimensions, 4 communication styles are distributed. A passive communication style refers to putting the wishes and needs of others before one's own, which can be frustrating. The person never speaks openly and honestly about what he thinks and feels, but unpleasant feelings accumulate in him, such as anger, rage, feeling of inferiority and Dr. Passive-aggressive communication style refers to the indirect expression of one's own desires and needs, but in such a way that the person avoids responsibility. Aggressive communication style refers to putting one's own desires and needs first, not respecting the desires and needs of others. This is an inadequate style of communication where the focus is on "winning" others. A person with an aggressive communication style instantly feels powerful and relieves their own tension, but these effects are short-term. In the long run, this communication style can lead to damaged relationships with other people who may feel hurt, angry, and humiliated. Therefore, it is easy to conclude that an assertive style of communication brings the most benefits to us, as well as to others. It enables the satisfaction of needs because we openly, honestly, and clearly express our own attitudes, opinions, and feelings, and at the same time we also respect the attitudes, opinions and feelings of other people. We express ourselves more clearly, so it is more likely that other people understand us better. We stand up for ourselves and what we do, and thus we are more satisfied with ourselves. Assertiveness allows us to deal with criticism and conflicts more efficiently, and at the same time it makes it easier for us to deal with difficult situations and relationships. Using an assertive style of communication, we achieve honest and open friendly, partner and business relationships. Finally, we improve our self-image.

3.2. Communication skills

Communication skills are skills that make it possible to achieve effective communication. There are numerous communication skills that would be desirable to master, but the key ones are active listening, paraphrasing, summarizing, and asking open questions, receiving, and giving quality feedback and the already described assertiveness.

Communication can be verbal and non-verbal. Verbal refers to every message that we speak and/or write, that is, to those messages that we try to convey using words. Non-verbal communication refers to facial expressions, tone and pitch of voice, speech, and body position, etc. Non-verbal communication complements verbal communication in such a way that it gives it an emotional tone, so we can interpret the sent message more easily and precisely. When there is a lack of non-verbal communication (e.g. using SMS or social networks) then it is possible to completely misinterpret the sent message, which can lead to conflict and even damage to the relationship.

Co-funded by the Erasmus+ Programme of the European Union

3.3. Development of social skills

Social skills are acquired spontaneously from early childhood, primarily in the family. The methods by which early social skills are acquired are imitation and the method of trial and error. As the child grows, he interacts with other people and groups (example: in kindergarten with peers and teachers, parents of other children, etc.). Through these interactions, social skills are established or changed, and the method by which they are most often adopted is model learning.

During adolescence, social interactions with peers are most common, and "the need to belong is a strong motive for joining peer groups" (group of authors, 2010). Interactions with peers are important for healthy cognitive, emotional, and social development. To achieve as many social interactions as possible, adolescents join social networks.

In adulthood, social interactions with people from childhood and adolescence are maintained through social networks or are extinguished. However, the social interactions that are realized by continuing education or starting a professional career are increasing. The social skills acquired until then are applied or adapted and upgraded in the new social context.

In old age, the number of social interactions decreases due to the reduction of living space, communication with a smaller number of people and a smaller number of social roles. However, it is important to emphasize that research has shown that the quality of close relationships is the most important factor for a sense of well-being and satisfaction with life.

4. Impact of technology on social skills

Considering the described historical development of technology, it is noticeable that the first purpose of technology was to enable survival, and then to facilitate and speed up production. Technology has "crept" into various areas of human activity, and social skills themselves have begun to be affected by the development of various communication systems. The goal was (but still is) to convey a clear and unambiguous message.
With the development of the Internet, mobile telephony, social networks and communication platforms, an extremely large number of people have access to various information from all over the world, but also the possibility of communicating and maintaining (close) relationships with many other people. The first studies showed that people who use the Internet more often communicate less often with family members, have a smaller social network of people they are close to, and are lonelier and more depressed (Kraut et al., according to Antolović, 2013). The researchers explained the results by saying that the time people devote to using the Internet is a substitute for the time they spent in social interactions in an offline environment. Another explanation is that people use the Internet to replace higher quality social connections with lower quality ones. Repeated research gave completely opposite results. In other words, it has been shown that people who use the Internet more often have a larger circle of close friends and acquaintances, interact more often with friends and family members, are more involved in community activities, and are more skilled at using computers.

As people became more proficient with computers and experienced in using the Internet, the negative impacts of using the Internet disappeared. Such results were probably obtained because

Co-funded by the Erasmus+ Programme of the European Union

the number of Internet users has increased, so the social network in the online environment has also increased. Research has also examined the connection between extraversion, social support, and Internet use. The results indicate that more sociable people and those who have a high level of social support have greater social benefits. However, introverts could also benefit because they can use new communication options to create new connections with people and receive support and useful information.

However, the most common Internet users are adolescents and young adults, mostly because communication on the Internet does not take place in real time, but gives them the time needed to think correct, and edit messages. In other words, it gives them control over social interactions and enables better self-presentation. However, it should be considered that in communication over the Internet, verbal communication is most often used, that is, there is a lack of non-verbal communication that gives the emotional tone of the message, so misunderstandings are possible due to misinterpreted messages. These risks are further increased when using social networks. Namely, a very common phenomenon is cyberbullying, which refers to the presentation of false, unpleasant, or hostile information about another person. Sexting, which refers to sending or publishing photos or videos in which a person is naked or half-naked, is also a frequent phenomenon. The so-called also appears. Facebook depression that occurs due to comparing yourself and your life with photos of others and illusions of their better lives. Many people still do not understand that everything they post or do in the online environment leaves a "digital footprint" that can harm them in the future.

However, it is important to emphasize that the use of the Internet, as well as social networks, has its advantages. Namely, social networks provide us with the possibility of connecting with family and friends, making new friendships, but also exchanging ideas, developing creativity, participating in various humanitarian activities, becoming active in the local community, and meeting others with whom we share the same interests, whom we would have difficulty getting to know in offline environment. They also enable the gathering of students outside of school, but also the exchange of ideas about different tasks or joint work on a group project.

5. Interactive methods for the development of social skills

Social skills are developed from an early age by the method of imitation (the child repeats the action but does not know what it is for or what the consequences are) and by the method of trial and error. Therefore, at that age, the most important interaction with the child is important people (parents, grandparents, educators) who should be aware that the child is watching them and repeating what they do. Interaction with peers is also important for children because in these interactions the child learns how to deal with conflict situations and how to fight for himself. Early social skills are acquired in these ways. At a slightly older age, children begin to learn according to the model, which is still most often parents, but it can also be teachers. By learning by model, the child also considers the consequence of a particular behavior, so if it is positive for the model, it is more likely that he will repeat it, and if it is negative, he will try to avoid it. Research has shown that the models that children watch in an offline environment have a greater influence than when the model shows them certain behaviors in the online environment. Adolescents are the biggest users of digital technologies, but they still learn according to the model, so the same as mentioned above would apply to them as well. However, precisely because of the large amount of time spent using digital technologies, adolescents could practice certain social skills. For example, how to write a unique and clear

message, how to use I-messages and be assertive, etc. These skills could be mastered by using the so-called programmed learning. Programmed learning refers to a system that contains all the necessary knowledge that is to be imparted to the student, but they are not offered immediately and all, but the student masters the content at his own pace. At the same time, after smaller units, the student's understanding is checked. In case of understanding, the student continues learning, and in case of misunderstanding, he goes back. In such a system, it is possible to simulate different social situations in which the student was asked to answer how to react or what should be done, how to answer, etc. In late adolescence and early adulthood, the social network changes or for continuing education or for starting professional development. At that age, new social skills could be mastered by watching videos, tutorials, etc. People at that age are mature enough to understand and can repeat and practice the behaviors shown. They too could establish or change social skills as previously described through programmed learning.

6. Conclusion

We do not need or can prevent the development of technology, but what we can do is adapt better and work on education, especially of children and young people, to maintain a symbiotic relationship. In other words, the use of technology for which we are not timely and adequately educated can lead to technology taking control of our lives, which can have far-reaching consequences on our mental health, educational and professional development, and interpersonal relationships. Therefore, it is recommended, along with education and moderation, because technology (at least in this form) still cannot replace the development of social skills that are realized in an offline environment. However, responsible use of technology can contribute to improving the quality of existing relationships, establishing new relationships, developing creativity, exchanging ideas, encouraging activities in an offline environment, etc.

Co-funded by the
Erasmus+ Programme
of the European Union

Bibliography

1. https://dictionary.apa.org/

2. https://fredcavazza.net/2022/06/14/panorama-des-medias-sociaux-2022/#more-38456

3. https://www.shutterstock.com/image-vector/social-skills-we-use-everyday-interact-1933402433

4. Antolović, A. (2013): „Kako korištenje Interneta utječe na socijalne interakcije ljudi?“, http://www.istrazime.com/socijalna-psihologija/kako-koristenje-interneta-utjece-na-socijalne-interakcije-ljudi/, Portal i udruga Istraži Me, 8.10.2022.

5. Berk, L. E. (2008): „Psihologija cjeloživotnog razvoja“, Jastrebarsko, Naklada Slap

6. Fuller, J. R. (2015): „Them Impact Of Social Media Use On Social Skills“, https://www.newyorkbehavioralhealth.com/the-impact-of-social-media-use-on-social-skills/, New York Behavioral Health, 8.10.2022.

7. Grbavac, J. i Grbavac, V. (2014): „Pojava društvenih mreža kao globalnog komunikacijskog fenomena“, Media, culture and public relations, 5, 2014, 2, str. 206-219 (https://hrcak.srce.hr/file/188969)

8. Skupina autora, (2010): „Trebamo li brinuti?“, https://www.poliklinika-djeca.hr/publikacije/trebamo-li-brinuti/, Poliklinika za zaštitu djece i mladih Grada Zagreba, 8.10.2022.

9. Župan, B., (2017): „Povijest i tehnologija: ono što ne znamo važnije je od onog što znamo“, https://gkr.hr/Lab/Citaonica/Povijest-i-tehnologija-ono-sto-ne-znamo-vaznije-je-od-onog-sto-znamo, Gradska knjižnica Rijeka, 8.10.2022.Dyer, Gregersen, and Christensen - The Innovator’s DNA: Mastering the Five Skills of Disruptive Innovators

10. Harkavy Daniel - Cele 7 perspective ale liderilor eficienti Metode testate pentru decizii mai bune si o influenta mai mare, Amaltea, 2021

11. Roxana Carmen Cordoş, Bacali, L. şi alţii , Antreprenoriat, Ed. Todesco, Cluj-Napoca, 2008

12. Stanier Bungay Mickael, Coaching zi de zi, ed. Publica, 2017

Co-funded by the Erasmus+ Programme of the European Union

II. Interactive methods to develop innovative skills

1. Discover my element

KEY WORDS: skills, career choice, talent, personal development, recognizing own interests

YOU WILL BE ABLE TO:

- develop awareness of their skills and capablities
- better identify their interests and values
- develop self-awareness
- define life goals and skills
- empower themselves and recognize the importance of empowering others.

DISCUSSION QUESTIONS:

- How easily did you find your element? Do you have more than one?
- What is your conclusion?
- If you had to choose between what you are good at and what you love, what would you pick?
- How many of you would choose a career only because of easier employment or better salary?
- Will you follow your parents' advice or your own interests and why?
- Which careers exist today that your parents haven't heard of?
- Where could your skills and talents lie? How would you group people's activities and careers?

Write a few things into each column (things you love, things you are good at, things you are keen on doing and things you have support for), and find out where they overlap. What appears in most columns might be your element!

Co-funded by the Erasmus+ Programme of the European Union

I love...	I am good...	I am keen on...	I need help with...	I have an opportunity to...

The concept of 'element' is based on the book The Element: How Finding Your Passion Changes Everything by Sir Ken Robinson (2010). The element is the point at which natural talent meets personal passion. It is here that people feel most themselves, inspired and able to achieve at their highest levels.

Holland's career typology

Realistic – working with tools, products, food, mechanical objects, technical drawings, plants, animals, have physical skills.

Investigative – understanding and solving scientific and mathematical problems

Artistic – theatre, painting, dance, music, writing, handicrafts

Social – helping others, teaching, caregiving, providing information

Enterprising – managing, persuading, selling ideas and products

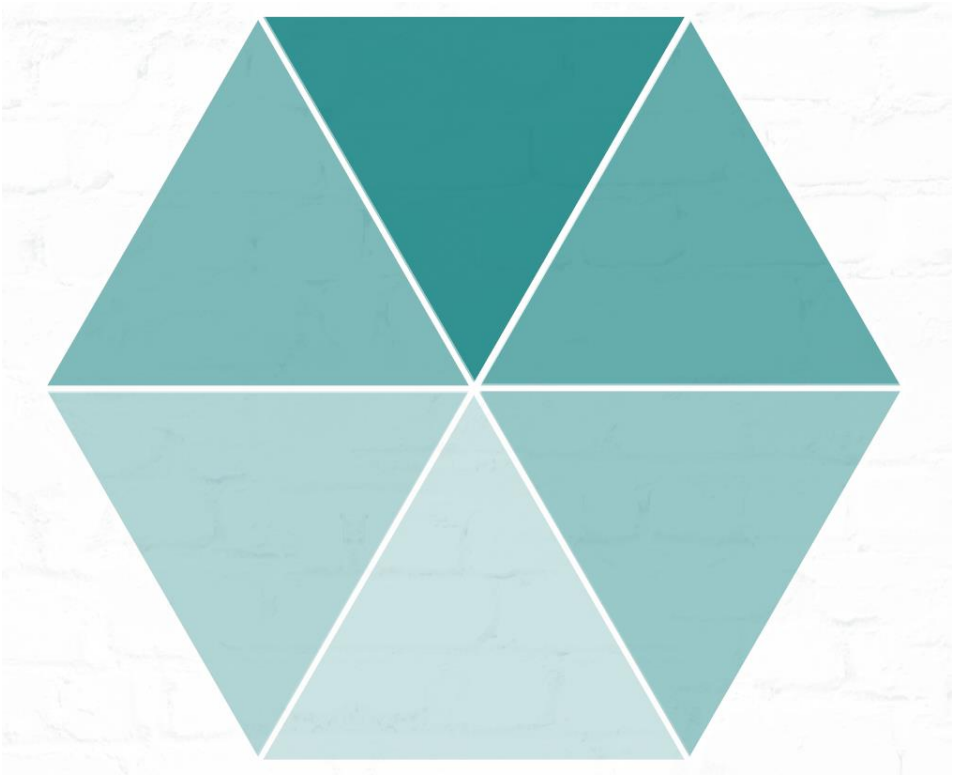
Conventional – working with numbers, ata, organization and editing

Co-funded by the Erasmus+ Programme of the European Union

How can we help young people find their ‘spark ‘or calling in life?
By: 1. recognizing and appreciating their ‘spark’; 2. creating possibilities for trying out different activities. 3. listening carefully when they are talking about what interests them. 4. showing support. 5. acting as a role-model and sharing our interests and ‘sparks’ and 6. including these activities in the curriculum. (William Damon: The Path to Purpose: Helping Our Children Find Their Calling in Life. 2008.)

Holland’s hexagram

In each part of the hexagram write down skills and experiences you have in that area:
1. Realistic/Doers: what are you skillful at?
2. Investigative/Thinkers: what kind of problems do you like solving?
3. Artistic: what do you like creating and how do you express yourself creatively?
4. Social: Who and how do you like helping?
5. Enterprising: which things do you start on your own, and others follow you?
6. Conventional/Organizers: In what ways are you tidy and well organized?



Take into consideration how important each of these activities is to you and how much you like it.

Co-funded by the Erasmus+ Programme of the European Union

Then try to choose up to three areas you feel are your strong suit. Which career would enable you to show all your qualities?

The Holland Hexagram is widely used to help people understand themselves and guide their choice of career. It was developed by American psychologist John L. Holland (1959).

2. From inspiration to action

KEY WORDS: interview, encouragement, giving support, mentoring

YOU WILL BE ABLE TO:

- get to know each other better
- practice active listening
- improve interviewing skills
- develop awareness of their own goals and choose priorities
- learn to plan.

DISCUSSION QUESTIONS:

- How did you like working in pairs?
- What was it like asking questions, and what was like answering them and talking about yourself?
- What did you realize and learn?
- Do you think it is important to plan or just let things run their course? When is it desirable to do one way, and when the other?
- What can you do when you are not good at something and you have problems reaching your goal?

FOUR INTERVIEW QUESTIONS:

1. What is your goal now? Choose one you would like to talk about today.
2. Imagine your ideal future! Keep imagining! How do you see yourself after achieving your goal? What exactly does it look like?
3. How are you getting there? What is your way of accomplishing it?
4. What will you do first (starting today or tomorrow) and who will help you?

Another way to start this workshop (depending on the amount of time you have) is the activity called The river of learning. This is a creative activity where you draw and color a river which represents their life from birth until now on a large sheet of paper. Then, students use symbols, illustrations, or words to mark important knowledge or skills they acquired, crucial moments and people they learned the most from.

Bibliography

Ana Munivrana, Domagoj Moric, Eli Pijaca Plavšic, Mario Bajkuša, Milana Rastovic, Vanja Kožic: workshop collection ''start the change!'', ACT Printlab, Zagreb 2021

Co-funded by the Erasmus+ Programme of the European Union

Co-funded by the
Erasmus+ Programme
of the European Union

ROMUAS
KILLS
RDUINO
SING
MACHINES
OUTDATED
ECOVER

Entrepreneurial skills

C4

Expert program toolkit

PR1

4) Entrepreneurial skills

I. Theoretical part

1. General consideration
2. The characteristics and skills of an entrepreneur
    1. What Are Entrepreneurial Skills?
    2. The Personal Characteristics of an Entrepreneur
    3. Entrepreneurial Interpersonal Skills
    4. Critical and Creative-Thinking Skills for Entrepreneurs
    5. Practical Entrepreneurial Skills and Knowledge
3. Description of the common skills of an entrepreneur
    1. Leadership skills
    2. Team management
    3. Problem solving
    4. Decision making
    5. Communication skills
    6. Creativity and innovation
    7. Time management
    8. Risk management
4. Is Entrepreneurship Right for you?
5. Bibliography

II. Interactive methods to develop entrepreneurial skills

1. Methods of stimulate the creativity of entrepreneurs
2. Interactive methods to develop entrepreneurial skills
    1. Are you ready to be an entrepreneur?
    2. Start your business with a paper clip
    3. Business Model Canvas – Plan your business in one page!
    4. Run your organization in style
3. Bibliography

Co-funded by the Erasmus+ Programme of the European Union

I. Theoretical part

1. General consideration

Entrepreneurship requires a unique set of skills including self-determination, self-motivation and perseverance. Whether you've conceptualized a solution to a known problem or you've simply decided to work for yourself, you'll want to establish a business to cement your online presence from the start.
Entrepreneurship is the act of starting a business in the hope of earning a profit. However, our modern perception of entrepreneurship has evolved into recognizing its ability to solve large-scale problems and influence social change. Entrepreneurs may also become thought leaders in their fields, although this isn't necessarily a distinguishing factor of someone in this role, but it can be a natural outcome.

Entrepreneurship also plays an important role in the economic development of our vibrant marketplace. It leads to better standards of living and generates new wealth, as well as increased employment and national income. Entrepreneurs also have the opportunity to help bridge the digital divide, the gap between those with access to internet and those without.

In the 2000s, entrepreneurship was extended from its origins in for-profit businesses to include social entrepreneurship, in which business goals are sought alongside social, environmental or humanitarian goals and even the concept of the political entrepreneurship. Entrepreneurship within an existing firm or large organization has been referred to as intrapreneurship and may include corporate ventures where large entities "spin-off" subsidiary organizations.[1]
Entrepreneurs are leaders willing to take risk and exercise initiative, taking advantage of market opportunities by planning, organizing and deploying resources,[2] often by innovating to create new or improving existing products or services.[3] In the 2000s, the term "entrepreneurship" has been extended to include a specific mindset resulting in entrepreneurial initiatives, e.g. in the form of social entrepreneurship, political entrepreneurship or knowledge entrepreneurship.

1. *Shane, Scott Andrew (2000). A General Theory of Entrepreneurship: The Individual-opportunity Nexus. Edward Elgar Publishing. ISBN 978-1-78100-799-0.*
2. Deakins & Freel 2009
3. Johnson, D. P. M. (2005). "A Glossary of Political Economy Terms, 2005". Auburn University. Archived from the original on 20 July 2014.

2. The characteristics and skills of an entrepreneur

With technological advancements and market demand for new business ideas and innovation, the practice of entrepreneurship is growing rapidly. Today, more and more self-motivated individuals are looking to become fully independent in their professional lives and are aspiring to create their own future.

1. What Are Entrepreneurial Skills?

Entrepreneurial skills are those normally associated with being an entrepreneur, although anyone can develop them.
Being an entrepreneur usually means starting and building your own successful business, but people with entrepreneurial skills can thrive within larger organizations, too.

Co-funded by the Erasmus+ Programme of the European Union

Many researchers have studied entrepreneurial skills, but found no definitive answers. Some common themes are:

- Personal characteristics.
- Interpersonal skills.
- Critical and creative-thinking skills.
- Practical skills and knowledge.

Regardless of how you define it, entrepreneurship isn't easy.
The following sections examine each skill area in more detail, and look at some of the questions you'll need to ask yourself if you want to become a successful entrepreneur.

2. The Personal Characteristics of an Entrepreneur

Do you have the mindset to be a successful entrepreneur? For example, entrepreneurs tend to be strongly innovative in outlook, and they may take risks that others would avoid.
Examine your own personal characteristics, values and beliefs, and ask yourself these questions:

- **Optimism:** Are you an optimistic thinker? Optimism is an asset, and it will help you through the tough times that many entrepreneurs experience as they find a business model that works for them.

- **Initiative:** Do you have initiative, and instinctively start problem-solving or business-improvement projects?

- **Drive and persistence:** Are you self-motivated and energetic? And are you prepared to work hard, for a very long time, to realize your goals?

- **Risk tolerance:** Are you able to take risks, and make decisions when facts are uncertain?

- **Resilience:** Are you resilient, so that you can pick yourself up when things don't go as planned? And do you learn and grow from your mistakes and failures?

3. Entrepreneurial Interpersonal Skills

As an entrepreneur, you'll likely have to work closely with others – so it's essential that you're able to build good relationships with your team, customers, suppliers, shareholders, investors, and other stakeholders.
Some people are more gifted in this area than others, but you can learn and improve these skills.
The types of interpersonal skills you'll need include:

- **Leadership and motivation:** Can you lead and motivate others to follow you and deliver your vision? And are you able to **delegate** work to other people? As an entrepreneur, you'll have to depend on others to get beyond the early stages of your business – there's just too much to do by yourself!
- **Communication skills:** Are you skilled in all types of communication? You need to be able to communicate well to sell your vision of the future to a wide variety of audiences, including investors, potential clients and team members.

Co-funded by the Erasmus+ Programme of the European Union

- **Listening:** Do you hear what others are telling you? Your ability to listen and absorb information and opinions can make or break you as an entrepreneur. Make sure that you're skilled at active and **empathic listening** .
- **Personal relationships:** Do you have good "people skills"? Are you self-aware, good at regulating your emotions, and able to respond positively to feedback or criticism?
- **Negotiation:** Are you a strong negotiator? Not only do you need to negotiate favorable prices, but you'll also need to resolve differences between people in a positive, mutually beneficial way.
- **Ethics:** Do you deal with people based on respect, integrity, fairness, and trust? Can you lead ethically? You'll find it difficult to build a happy, productive business if you deal with staff, customers or suppliers in a shabby way.

4. Critical and Creative-Thinking Skills for Entrepreneurs

As an entrepreneur, you need to come up with fresh ideas, and make good decisions about opportunities and potential projects.
Many people think that you're either born creative or you're not. But creativity is a skill that you can develop, and there are many tools available to inspire you.

- **Creative thinking:** Are you able to see situations from a variety of perspectives to generate original ideas?
- **Problem solving:** You'll need sound strategies for solving business problems that will inevitably arise.
- **Recognizing opportunities:** Do you recognize opportunities when they present themselves? Can you spot a trend? And are you able to create a workable plan to take advantage of the opportunities you identify?

5. Practical Entrepreneurial Skills and Knowledge

Entrepreneurs also need solid practical skills and knowledge to produce goods or services effectively, and to run a company.

- **Goal setting:** Setting SMART goals (Specific, Measurable, Achievable, Relevant, and Time-Bound) will focus your efforts and allow you to use your time and resources more effectively.
- **Planning and organizing:** Do you have the talents, skills and abilities necessary to achieve your goals? Can you coordinate people to achieve these efficiently and effectively? Strong project-management skills are important, as are basic organization skills. And you'll need a coherent, well thought-out business plan , and the appropriate financial forecasts .
- **Decision making:** Your business decisions should be based on good information, evidence, and weighing up the potential consequences. Core decision-making tools include Decision Tree Analysis, Grid Analysis, and Six Thinking Hats .

You need knowledge in many different areas when you're starting or running a business, so be prepared for some serious learning!

Co-funded by the Erasmus+ Programme of the European Union

Be sure to include:

- **Business knowledge:** Ensure that you have a working knowledge of the main functional areas of a business: sales, marketing, finance, and operations. If you can't fulfill all these functions yourself, you'll need to hire others to work with you, and manage them competently.
- **Entrepreneurial knowledge:** How will you fund your business, and how much capital do you need to raise? Finding a business model that works for you can require a long period of experimentation and hard work.
- **Opportunity-Specific Knowledge:** Do you understand the market you're attempting to enter, and do you know what you need to do to bring your product or service to market?
- **Venture-Specific Knowledge:** Do you know what it takes to make this type of business successful? And do you understand the specifics of the business that you want to start?
  You can also learn from others who've worked on projects similar to the ones that you're contemplating, or find a mentor – someone else who's been there before and is willing to coach you.

Co-funded by the
Erasmus+ Programme
of the European Union

3. Description of the common skills of an entrepreneur

1. Leadership skills

There is no unique way to describe **great leadership skills**. Unsurprisingly, you can find different lists describing the leadership skills one should possess to become a great leader in the workplace.
In essence, **leadership skills** are the abilities people have to lead and deliver projects, encourage initiatives, build a sense of common purpose, and empower others.
Leadership skills also include the abilities people have to steer employees toward the achievement of the business goals, inspire them, drive change, and deliver results.

Source: https://www.thebalancecareers.com/top-leadership-skills-2063782

2. Team management

While there is no single metric or leadership skill that define a great team manager, there are several defining aspects of a well-working team that showcase strong leadership.
Successful management is when everyone has:

- A clear understanding of their own responsibilities
- A desire to achieve team and organizational goals
- Regular check-ins by managers and supervisors

An excellent manager also makes time to listen to and carry out employee feedback as needed, lead by example—whether that's through showing a good work life balance or getting tasks done on time—and stay transparent on their own work challenges and how they'll solve them.

Co-funded by the Erasmus+ Programme of the European Union

Source: https://www.marketing91.com/team-management/

3. Problem solving

Problem solving is the way by which solutions are developed to remove an obstacle from achieving an ultimate goal. Problems in need of solutions range from basic personal issues of "how do I turn on this appliance?' to more complex topics in the business and academic fields. The former scenario falls into the category of simple problem solving (SPS), whereas the latter is known as complex problem solving (CPS). Simple problems have one issue, while complex problems have multiple obstacles that simultaneously need solutions. [1]

1. Frensch, Peter A.; Funke, Joachim, eds. (2014-04-04). *Complex Problem Solving*. doi:10.4324/9781315806723. ISBN 9781315806723.

https://www.youtube.com/watch?v=klAE9ML0XLs

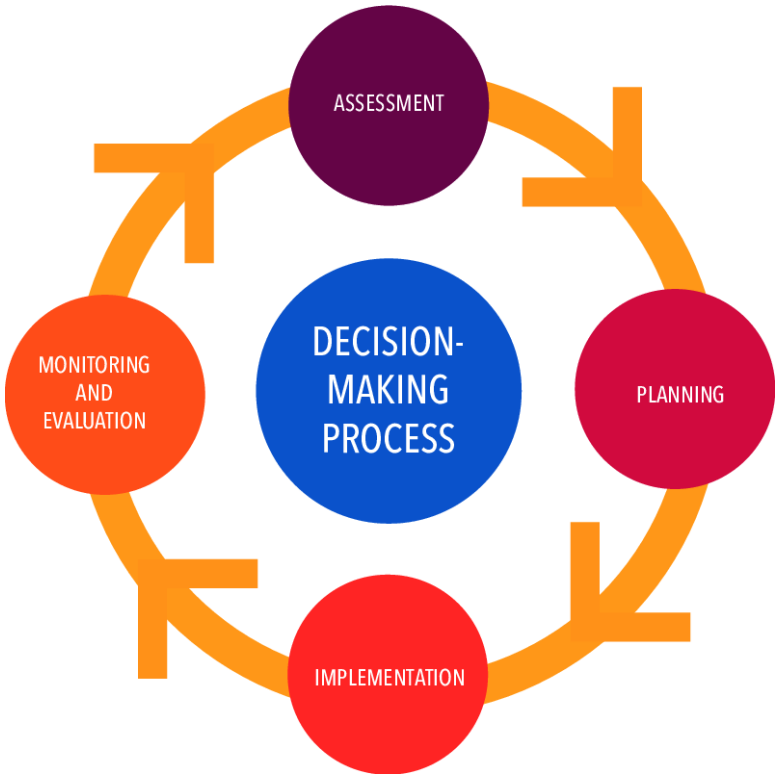
4. Decision making

Decision making is **the process of making choices by identifying a decision, gathering information, and assessing alternative resolutions**. Using a step-by-step decision-making process can help you make more deliberate, thoughtful decisions by organizing relevant information and defining alternatives.

We describe the sequence of steps involved in the decision-making process.

Co-funded by the Erasmus+ Programme of the European Union

1. ***Classifying the problem.*** Is it generic? Is it exceptional and unique? Or is it the first manifestation of a new genus for which a rule has yet to be developed?
2. ***Defining the problem.*** What are we dealing with?
3. ***Specifying the answer to the problem.*** What are the "boundary conditions"?
4. ***Deciding what is "right," rather than what is acceptable, in order to meet the boundary conditions.***. What will fully satisfy the specifications *before* attention is given to the compromises, adaptations, and concessions needed to make the decision acceptable?
5. ***Building into the decision the action to carry it out.*** What does the action commitment have to be? Who has to know about it?
6. ***Testing the validity and effectiveness of the decision against the actual course of events.*** How is the decision being carried out? Are the assumptions on which it is based appropriate or obsolete?



Source: https://www.researchgate.net/figure/Decision-making-for-ecosystem-service-management_fig2_313161530

5. Communication skills

Communication skills involve **listening, speaking, observing and empathizing**. It is also helpful to understand the differences in how to communicate through face-to-face interactions, phone conversations and digital communications, like email and social media.
The seven C's of communication are a list of principles for written and spoken communications to ensure that they are effective. The seven C's are: **clarity, correctness, conciseness, courtesy, concreteness, consideration and completeness**.

Co-funded by the Erasmus+ Programme of the European Union

Source: https://www.thebalancecareers.com/communication-skills-list-2063779

6. Creativity and innovation

Creativity is the ability to transcend traditional ways of thinking or acting, and to develop new and original ideas, methods or objects.
Innovation is the practical implementation of ideas that result in the introduction of new goods or services or improvement in offering goods or services.
Creativity is the act of turning new and imaginative ideas into reality. Creativity is characterized by the ability to perceive the world in new ways, to find hidden patterns, to make connections between seemingly unrelated phenomena, and to generate solutions. Creativity involves two processes: thinking, then producing.
Creativity begins with a foundation of knowledge, learning a discipline, and mastering a way of thinking. You can learn to be creative by experimenting, exploring, questioning assumptions, using imagination and synthesing information.
Studies by Clayton M. Christensen and his researchers uncovered The Innovators DNA**:** Your ability to generate innovative ideas is not merely a function of the mind, but also a function of five key behaviors that optimize your brain for discovery:

1. **Associating:** drawing connections between questions, problems, or ideas from unrelated fields
2. **Questioning:** posing queries that challenge common wisdom
3. **Observing:** scrutinizing the behavior of customers, suppliers, and competitors to identify new ways of doing things
4. **Networking:** meeting people with different ideas and perspectives
5. **Experimenting:** constructing interactive experiences and provoking unorthodox responses to see what insights emerge

Co-funded by the Erasmus+ Programme of the European Union

Source: https://weareive.org/what-is-the-difference-between-creativity-and-innovation/

7. Time management

Time management is **the process of organizing and planning how to divide your time between different activities**. Get it right, and you'll end up working smarter, not harder, to get more done in less time – even when time is tight and pressures are high. The highest achievers manage their time exceptionally well.
https://www.youtube.com/watch?v=xgp6eELYY1M

8. Risk management

Risk management is **the process of identifying, assessing and controlling threats to an organization's capital and earnings**. These risks stem from a variety of sources including financial uncertainties, legal liabilities, technology issues, strategic management errors, accidents and natural disasters.
Risk management has perhaps never been more important than it is now. The risks modern organizations face have grown more complex, fueled by the rapid pace of globalization. New risks are constantly emerging, often related to and generated by the now-pervasive use of digital technology. Climate change has been dubbed a "threat multiplier" by risk experts.
https://www.youtube.com/watch?v=TcKoUe8vRE0

Co-funded by the Erasmus+ Programme of the European Union

Source: https://aipc.ae/risk-management-process/

4. Is Entrepreneurship Right for You?

Before you proceed with your plan to become an entrepreneur, assess your skills against all of the questions and considerations above. Use a Personal SWOT Analysis to examine your Strengths and Weaknesses, your Opportunities, and the Threats that you may face.

Be honest with yourself about your motivations and the level of commitment you're prepared to give to your project. This could prevent you from making a costly mistake.

As you work through your analysis, you may feel that you're ready to plunge into your exciting new venture. Alternatively, you may decide to wait and further develop your skills. You may even decide that entrepreneurship isn't for you after all.

Becoming an entrepreneur is an important career decision, so avoid the temptation to act impulsively. Do your homework. Reflect on your needs, your objectives, and your financial and personal circumstances. Entrepreneurialism can take a huge amount of time and dedication, so make sure that it feels right.

Co-funded by the Erasmus+ Programme of the European Union

5. Bibliography

1. https://www.investopedia.com/terms/e/entrepreneur.asp#:~:text=Entrepreneurship%20is%20when%20an%20individual,with%20an%20innovative%20new%20idea
2. https://en.wikipedia.org/
3. https://www.mindtools.com/pages/article/newCDV_76.htm
4. Anderson and Adams, “Mastering Leadership,” Wiley, 2015.
5. Bucur Viorel, Curs Antreprenoriat, ClujNapoca 2011
6. Dorel Ursu -Crearea şi dezvoltarea afacerilor –planul de afaceri, Modul 4, Programul “Investeşte în oameni”, 2007-2013
7. Dyer, Gregersen, and Christensen - The Innovator’s DNA: Mastering the Five Skills of Disruptive Innovators
8. Harkavy Daniel - Cele 7 perspective ale liderilor eficienti Metode testate pentru decizii mai bune si o influenta mai mare, Amaltea, 2021
9. Roxana Carmen Cordoş, Bacali, L. şi alţii , Antreprenoriat, Ed. Todesco, Cluj-Napoca, 2008
10. Stanier Bungay Mickael, Coaching zi de zi, ed. Publica, 2017

Co-funded by the Erasmus+ Programme of the European Union

II. Interactive methods to develop entrepreneurial skills

Creativity methods are used to find new product ideas. These methods do not seek to find solutions that meet the needs expressed on the market but seek to anticipate the evolution of consumer needs.
Creativity methods can be divided into two categories: methods
intuitive and rational methods.
Along with these methods we can add: The Delphi Group Method, the 6 Thinking Hats Method, The Lotus Technique (water lily flower), The Interdependent Groups Method (Mosaic), The Pyramid Method, Star bursting (stellar explosion), etc.
Intuitive methods are based on the idea that a group of individuals is much more creative than each individual working separately. Emphasis is placed on the synergistic effect resulting from the interaction between group members. Through these methods, the imagination and intuition of the people who participate in the creativity sessions are harnessed.
Rational methods involve the search for new product ideas, starting from the systematic analysis of existing products. A number of aspects can be taken into account in the analysis process, such as: the characteristics of the existing products, their behavior in consumption, methods of use, consumption situations, etc.

2.1 Are you ready to be an entrepreneur?

Preparation: 15 minutes
Duration: 40 minutes
Group size: 5-30 people
Age: 16+
Tags: entrepreneurship, self-evaluation
When is it used?
- assess the personal qualities/skills needed to become an entrepreneur
Materials:
• printed sheets with the entrepreneurial quotient test and the results
• pens and markers
• the printed explanations for the quiz
• flipchart sheet
How?

Preparation:
• Print the questionnaire for establishing the entrepreneurship quotient, adapted from Start Your Own Business, Sixth Edition: The Only Startup Book You'll Ever Need; the test was originally created by Northwestern Mutual Life insurance. You will need a copy of the test for each participant.
• Briefly read the presentation of the questionnaire in the same appendix

Presentation:

1. Ask the participants what, from their point of view, are the most important qualities of an entrepreneur. Write on a flipchart the things they mentioned. You can use a well-known entrepreneur in your community so that participants can have a real person in mind when thinking about specific qualities/skills.

Co-funded by the Erasmus+ Programme of the European Union

2. Once the brainstorming session is over, introduce the idea to the participants that entrepreneurial skills can be tested and that during this exercise they can test their own entrepreneurship quotient.
3. Give each participant a printed questionnaire and give them 15-20 minutes to answer the questions.
4. After everyone has finished completing the questionnaire, give each participant a printed sheet with the results of the questionnaire. Help participants rate their questionnaire accordingly by taking each question in turn. At the same time, it explains for each question why some answers are more "entrepreneurial" than others
5. Recap the most important skills people should develop to be successful entrepreneurs. Write down the new skills on the flipchart you used originally.
6. Remind the participants of the following: the fact that they have these qualities / skills, does not automatically mean that they will be successful in their business; also, someone who does not have these qualities/skills at the moment, in order to be successful can develop these skills over time; and in one case and another, it takes a lot of work to develop a business, skills alone are not enough.

Follow-up questions:
Now that you know what your entrepreneurship quotient is and if you are thinking of becoming an entrepreneur, what do you think are your skills that you can already rely on that will help you grow your business?
What skills do you need to develop and how do you plan to do this?
https://www.youtube.com/watch?v=lJjILQu2xM8

Source: 21stcenturyeducation.edu.pk/course/entrepreneurship/

Co-funded by the Erasmus+ Programme of the European Union

Appendix

Test your entrepreneurship quotient!

Adapted from Start Your Own Business, Sixth Edition: The Only Startup Book You'll Ever Need, by Inc the Staff of Entrepreneur Media; the test was originally developed by Northwestern Mutual Life insurance.
Before taking this test, remember: even if you have a high entrepreneurship quotient, it does not automatically mean that your business will be successful, but that you have the qualities/skills to be successful in a business that you need to work for much to develop it.
Don't think too much when answering the questions. Choose the first answer that comes to mind when you think of yourself. Start the test!
Answer the following questions with "YES" or "NO" and add up your score, as indicated in the section at the end of the test, to find out what your entrepreneurship quotient is.
YES NO

___ ___ 1. Are you one of the top students in your school?
___ ___ 2. Do you enjoy participating in group activities such as clubs or team sports?
___ ___ 3. Do you prefer to be alone?
___ ___ 4. Have you run for student council or started small business ventures, such as a lemonade stand, a class/family newspaper, or a card sale?
___ ___ 5. Are you stubborn?
___ ___ 6. Are you cautious?
___ ___ 7. Are you bold or adventurous?
___ ___ 8. Do other people's opinions matter a lot to you?
___ ___ 9. Would you start your own business just to escape your daily routine?
___ ___ 10. Even if you like to work, would you be willing to work overnight?
___ ___ 11. Are you willing to work no matter how long, even if you have had little or no sleep, to complete a project?
___ ___ 12. After successfully completing a project, do you start another right after?
___ ___ 13. Are you willing to use your savings to start a business?
___ ___ 14. Would you be willing to borrow money from someone else?
___ ___ 15. If your business failed, would you immediately start a new one?
___ ___ 16. Or would you start looking for a job with a regular salary?
___ ___ 17. Do you think it is risky to be an entrepreneur?
___ ___ 18. Do you write down your short-term or long-term goals?
___ ___ 19. Do you think you have the ability to manage your money well?
___ ___ 20. Do you get bored easily?
___ ___ 21. Are you optimistic?

Co-funded by the Erasmus+ Programme of the European Union

How to calculate your score:
1. If yes, subtract 4 points; if not, add 4.
2. If yes, subtract 1 point; if not, add 1.
3. If yes, collect 1 point; if not, subtract 1 .
4. If yes, collect 2 points; if not, subtract 2.
5. If yes, collect 1 point; if not, subtract 1.
6. If yes, subtract 4 points; if not, add 4.
7. If yes, collect 4 points; if not, you get no points.
8. If yes, subtract 1 point; if not, add 1.
9. If yes, add 2 points; if not, subtract 2.
10. If yes, add 2 points; if not, subtract 6.
11. If yes, collect 4 points; if not, you get no points.
12. If yes, collect 2 points; if not, subtract 2.
13. If yes, collect 2 points; if not, subtract 2.
14. If yes, collect 2 points; if not, subtract 2.
15. If yes, collect 4 points; if not, subtract 4.
16. If yes, subtract 1 point; if not, you get no points.
17. If yes, subtract 2 points; if not, add 2.
18. If yes, collect 1 point; if not, subtract 1.
19. If yes, collect 2 points; if not, subtract 2.
20. If yes, add 2 points; if not, subtract 2.
21. If 2, add 2 points; if not, subtract 2.

Test results
• A score of 35 points or more: Everything is going well for you. You have the potential to achieve spectacular entrepreneurial success.
• A score between 15 and 34: Your background, skills and talent give you an excellent chance to succeed in your business. You should go far.
• A score between 0 and 14: You have a good start in the ability and/or experience to run a business, and should be successful in starting one if you learn how to run it.
• A score between -15 to -1: You could succeed if you ventured out on your own, but you may need to work extra to compensate for the lack of advantages and skills that give entrepreneurs an edge when they start the business
• A score between -16 to -43: You probably have other talents than this one. You should think about whether building a business is what you really want to do, because you may find yourself swimming against the tide. Working in a company or for someone else, developing a career in a profession or gaining expertise in a technical field would be far more suitable for you and would allow you to enjoy a more fulfilling lifestyle. suitable for your skills and interests.

Interpretation:
If your score is high, it does not necessarily mean that you will be successful in your business, but that you have the necessary qualities to be successful in the business you started. If your score is low, it doesn't mean you won't succeed. There are other factors to consider before taking this step.

Co-funded by the Erasmus+ Programme of the European Union

2.2. Start your business with a paper clip

Preparation: 30 minutes
Duration: 45 minutes
Group size: 5-20 people
Age: 16+
Tags: entrepreneurship, creativity, alternative use, innovative thinking, "think out of the box"

When is it used?
• assessment of creativity potential;
• improvement of "out of the box" thinking / thinking skills;
• understanding how creativity and innovation help grow your business.
Materials:
• pens, post-its, flipchart sheets, markers, paper clips

How?
Preparation:
• Read the brief overview of the alternative uses test in the FACTS BOX.

Presentation:

1. Start the discussion by asking the participants for their definition of 'creativity' and write their answers on a flip chart.
2. Then ask them what are the main things / aspects that a person should have to be considered creative. Write these answers on the flipchart as well.
3. Give each participant a paper clip and lots of post-its, then give them 3 minutes to think of as many possible alternatives to using the paper clip. For each idea, they should use a different post-it.
4. Then ask the participants to stick post-its on the flipchart as they present them. Ask them to stick on the flipchart only the new ideas, the ones that have not been mentioned before.
5. After all the ideas are on the flipchart, divide the ideas on the post-its with the help of the participants into three categories:
    - Use the paperclip as it is;
    - Change the design of the paper clip;
    - Adds an object to the paperclip, creating another object.

Moreover, analyze the ideas together with the participants from the following perspectives:

- Fluency (how many uses each participant found compared to the total number of uses found by the group);
- Originality (how unusual is the idea);
- Flexibility (how many different sectors they cover with the ideas; for example, if someone mentions the idea of using paperclips as earrings but also as a necklace, then that person has only covered one sector – jewellery/accessories) ;
- Elaboration (how detailed the ideas were).

6. Ask participants how they came up with those ideas, and then introduce them to the idea that creativity can be trained through daily exercises like this one.
7. Now ask the participants why creativity might be important for a business. Based on their answers, explain to them that it is not enough to be creative, to think "out of the box", and that it is important to be able to apply creativity to solve problems in the community, to meet the

Co-funded by the Erasmus+ Programme of the European Union

needs of potential customers or business opportunities. Thus, creative ideas must be put into practice through innovation.

Follow-up questions:
Ask participants to say what happened once they saw other people's ideas about how to use a paper clip. Have they found any other ideas?
Ask them how many alternative uses they found before they really got creative with using a paper clip? What does that tell them? Help them learn that it is very important to accept challenges and that good ideas come later rather than early in the process.

FACTS BOX

Alternative Uses Test

Developed by J.P. Guilford in 1967, the Alternative Uses Task Test asks you to find as many uses as possible for an object, such as a brick, a shoe, a paper clip, etc., in a short period of time (usually between 2 and 5 minutes).
The test measures a person's divergent thinking skills. This contrasts with traditional tests that focus on convergent thinking, that is, a person's ability to find a single, "correct" solution to a problem. This test encourages people to come up with multiple solutions to a problem.
In various studies, the test showed that participants who found many ideas at a given time had much more creative answers, and the ideas offered at the beginning, among the first, were less creative compared to those in the second part of the list of ideas. In other words, creativity improves with practice and persistence.
The test evaluation includes four elements:
1. Fluency - how many uses a person can find;
2. Originality - how unusual the ideas are, a kind of proof of different thinking;
3. Flexibility – the variety of ideas, from various sectors or categories;
4. Elaboration - how detailed the ideas are described.

This test can be used every day, starting from various objects, as a tool for developing creativity.
https://www.youtube.com/watch?v=yjYrxcGSWX4
https://www.youtube.com/watch?v=tTtjETjGDFY

https://www.alamy.com/stock-photo-creative-creativity-innovation-design-vision-concept-126130669.html

2.3. Business Model Canvas – Plan your business in one page!

Preparation: 60 minutes
Duration: 90 minutes
Group size: 5-30 people
Age of participants: 16+
Tags: entrepreneurship, planning, business plan

When is it used?
• developing a business idea into a real plan, ready to be put into practice
• planning projects involving different types of clients and stakeholders

Materials:
flipchart sheets, post-it notes – of several colors, markers, video projector and laptop**Preparation**:
Watch the short video presentation about the business model canvas here: https://www.youtube.com/watch?v=QoAOzMTLP5s;

• Prepare a flipchart sheet, using the business model canvas available at this link: https://templatelab.com/business-model-templates/;

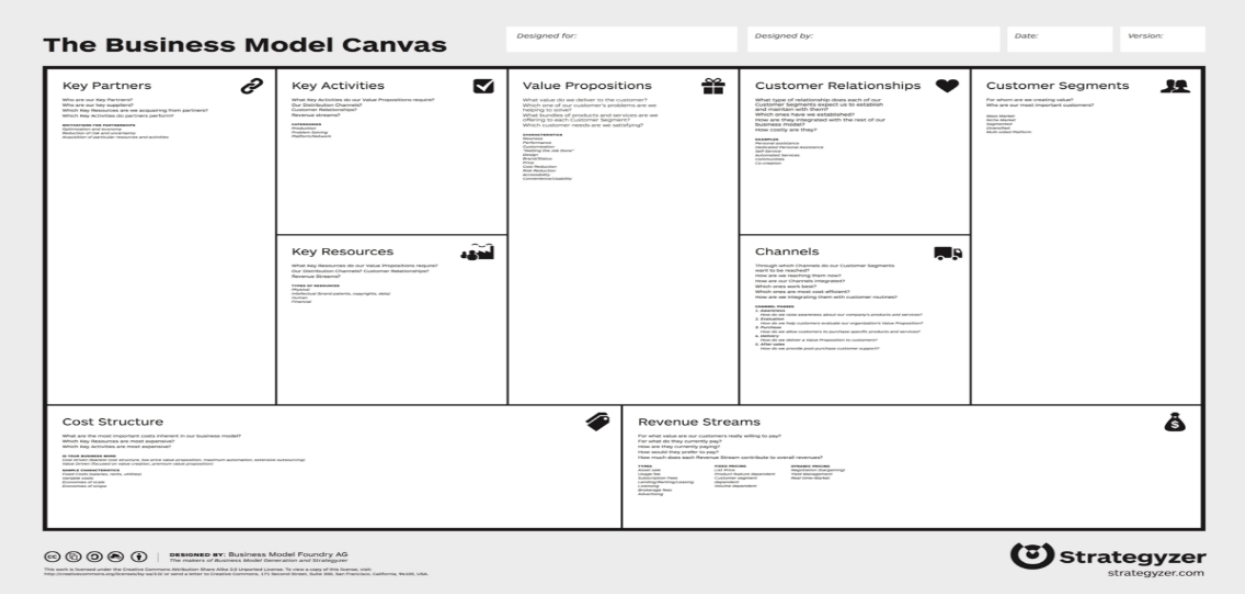
• Using multi-colored post-its, prepare the presentation of the project / business idea (choose a simple idea, maybe even familiar to the participants), following the nine sections in the canvas; use at least two types of customers so people understand how customers can become key partners or provide key resources to the project when another customer segment is discussed.

Presentation:

1. Ask the participants what are the main elements of any project (objectives, purpose, problem, beneficiaries, activities, methodology, estimated results, budget, etc.) or of any business idea (products, customers, distribution channels, prices, costs, promotion etc.).
2. It presents the business model canvas as a tool that has all the components of a project / business idea on a single page.
3. Exemplify the use of the business model canvas with the project / business idea you prepared before the workshop. It shows how the model works for at least two different customer segments. Show them how the nine elements are related to each other: customers / customer segments, examples of customer value (products / services), customer relationship, communication channels, revenue streams, key activities, key resources, key partners and cost structure.
4. Divide the participants into groups of five and give each a flipchart.
5. Ask each group to prepare a business model canvas on a flipchart sheet.
6. Ask each group to come up with a project idea that they would like to implement in their community / their school / their neighborhood and prepare a presentation of the project using the business model canvas.
7. Each group has between five and seven minutes to present their project.
8. Give feedback on every project idea. Don't focus on the idea of the project, but on how the project was planned using the business canvas model.

Co-funded by the Erasmus+ Programme of the European Union

Follow-up questions:
How did the project idea develop / change during the exercise?
What are the new aspects / ideas or new questions related to the project, generated with the use of the business model canvas?
What can you say about the sustainability of your project, looking at the business model you have developed?
https://www.youtube.com/watch?v=I8nwNcCfyig



Source: https://en.wikipedia.org/wiki/Business_Model_Canvas

2.4. Run your organization in style

Preparation: 25 minutes
Duration: 45 minutes
Group size: 15 people
Age: 16+
Tags: entrepreneurship, leadership, leadership styles

When is it used?
• learning and understanding different leadership styles
• understanding the advantages and disadvantages of each leadership style
Materials: flipchart sheets, markers

How?
Preparation:
• Print the three scenarios with the three types of leaders (democratic leader, autocratic leader and laissez-faire leader) from the appendix.

Co-funded by the Erasmus+ Programme of the European Union

Presentation:

1. Divide the participants into three groups and select one person from each group to be designated as leader.
2. Give each leader (but NOT the groups) one of the three scenarios in the appendix.
3. While the three leaders are reading their script, give each group a flipchart and markers and tell them they will receive instructions from their leaders.
4. Now let the groups be led by the leaders and solve the requirement in about 15 minutes.
5. Each group must present its decisions and the results of the group activity.

Follow-up questions:

What behaviors have you identified in your leaders? What about "followers"?
How did you feel as a leader or team member during the role play?
How effective was each leader-led meeting?
What were the advantages and disadvantages of each meeting determined by the leadership style?
https://www.youtube.com/watch?v=2lEp4TVpxgA

Source: https://www.indeed.com/career-advice/career-development/10-common-leadership-styles

Co-funded by the Erasmus+ Programme of the European Union

Appendix

Scenario A

You are a manager in a company with 250 employees. Top management asks you to make suggestions for organizing the annual party. The purpose of the party is to celebrate the good results of the company to which all employees have contributed, so you need to offer a wide variety of activities in your proposal. You have planned a brief meeting with your team. In the next 15 minutes you have to collect as many ideas as possible and come up with a plan for the party.
At the end of the meeting, one person will be nominated to present the results and answer questions.

Summary:

· Objective of the meeting: create an activity plan for the party.
· Duration: 15 minutes.
· Logistics: the plan will be written on a flip-chart sheet.
· One person will be nominated to present the results.

Try to follow the following behaviors well:

· You prepare carefully and take notes on the basic ideas you want to bring to the meeting.
· You are the only one who knows what is good for the team.
· You lead the discussion; gives orders to others about what needs to be done.
· Immediately assign tasks to each team member.
· Influence and dominate the discussion down to the last detail.
· You talk the most.
· You don't have to ask questions and listen; stop the conversation immediately if you feel it is unnecessary.
· Interrupt others often.
· Criticize and praise directly, openly.
· Do not ask team members to present their ideas or suggestions.
· You expect excellent results and judge your subordinates, considering yourself the best.
· After all, you already have the plan for the party and you just have to explain it to your subordinates.
· You must present the results.

Scenario B

You are a manager in a company with 250 employees. Top management asks you to make suggestions for organizing the annual party. The purpose of the party is to celebrate the good results of the company to which all employees have contributed, so you need to offer a wide variety of activities in your proposal. You have planned a brief meeting with your team. In the next 15 minutes you have to collect as many ideas as possible and come up with a plan for the party.
At the end of the meeting, one person will be nominated to present the results and answer questions.

Summary:

· Objective of the meeting: create an activity plan for the party.
· Duration: 15 minutes.
· Logistics: the plan will be written on a flip-chart sheet.

Co-funded by the Erasmus+ Programme of the European Union

· One person will be nominated to present the results.

Try to follow the following behaviors well:

· From the very beginning, you inform your team about the objectives of the meeting, what is expected to be achieved at the end and how to discuss it.
· Reach an understanding regarding the organization of the meeting and individual tasks.
· Ask team members to make specific suggestions and encourage everyone to be active.
· Ask questions for support and encouragement; listen carefully.
· You involve your colleagues in decision-making.
· You explain your own opinions to them and do not impose them.
· You involve the quietest and encourage everyone to contribute.
· Do not force anything, but support activities for all employees.
-Controls the discussion only when necessary and supports individual initiatives or leadership of the meeting by team members.

Scenario C

You are a manager in a company with 250 employees. Top management asks you to make suggestions for organizing the annual party. The purpose of the party is to celebrate the good results of the company to which all employees have contributed, so you need to offer a wide variety of activities in your proposal. You have planned a brief meeting with your team. In the next 15 minutes you have to collect as many ideas as possible and come up with a plan for the party.
At the end of the meeting, one person will be nominated to present the results and answer questions.

Summary:

· Objective of the meeting: create an activity plan for the party.
· Duration: 15 minutes.
· Logistics: the plan will be written on a flip-chart sheet.
· One person will be nominated to present the results.

Try to follow the following behaviors well:

· Begin by briefly informing your subordinates about the objectives of the meeting; without giving too many details; ask them to start working.
· Be more of an observer.
· Let them talk to each other and don't interfere.
· You are sure they are able to handle it without you getting involved.
· Do not express your opinions.
· Do not participate in discussions and avoid eye contact.
· Wait for questions to be addressed directly to you.
· Answer the questions with other questions or just ask them - "But what do you think?".
· Do not criticize, do not praise.
· You have a neutral role; you are not the team leader.
· Another person should present the answers.

Co-funded by the Erasmus+ Programme of the European Union

3. Bibliography

http://www.tydzienprzedsiebiorczosci.pl/files/best_practices_EN.pdf
Start Your Own Business, Sixth Edition: The Only Startup Book You'll Ever Need, written by Inc The Staff of Entrepreneur Media. http://www.amazon.com/Start-Your-Own-Business-Sixth
How to Improve Your Critical Thinking Skills and Make Better Business Decisions
http://www.entrepreneur.com/article/226484
Articol despre luarea deciziilor:
https://new.edu/resources/making-rational-decisions
Dan Ariely: Are we in control of our own decisions?, TED Talks
https://www.ted.com/talks/dan_ariely_asks_are_we_in_control_of_our_own_decisions?language=en
Cum să atragi investitorii: http://www.iseeyou-network.eu/files/8514/5579/7112/Business_Angel_Attraction_Plan.pdf
Robert Kiyosaki, *Rich Dad's Cashflow Quadrant. Guide to Financial Freedom* (2012).
Trump and Kiyosaki on Building Business Wealth, http://www.entrepreneur.com/article/169174
Business Model Canvas
http://www.businessmodelgeneration.com/
http://nonprofitcanvaskit.com/
The 7 habits of highly effective people, by Stephen Covey:
https://www.stephencovey.com/7habits/7habits-habit1.php
7 Reasons '7 Habits of Highly Effective People' Lives on 25 Years Later
http://www.entrepreneur.com/article/236544
Leadership Styles and Types: Authoritarian, Laissez-Faire & Democratic
http://study.com/academy/lesson/leadership-styles-and-types-authoritarian-laissez-faire-democratic.html
Leadership situațional: https://situational.com/the-cls-difference/situational-leadership-what-we-do/
Stilurile de leadership:
https://www.mindtools.com/pages/article/newLDR_84.html
Clutterbuck, David & Chartered Institute of Personnel and Development. *Everyone needs a mentor: fostering talent in your organisation*. (Chartered Institute of Personnel and Development, 2004).
Crețu D., Daniliuc F., Vrînceanu C. Ghid practic pentru antreprenori, Evrika Publishing, 2021
Goleman David. Leadership**.**Puterea inteligentei emotionale. Selectie de texte**, Curtea veche, București, 2016**
Humphrey, Ronald H. *Effective leadership: theory, cases, and applications*. (SAGE, 2014).

Co-funded by the Erasmus+ Programme of the European Union

Co-funded by the
Erasmus+ Programme
of the European Union

ROMUAS
KILLS
RDUINO
SING
MACHINES
OUTDATED
ECOVER

Green skills

C5

Expert program toolkit

PR1

5) Green skills

I. Theoretical part

1. Environmental sustainability and climate change

2. Towards a greener economy

3. What are Green Skills

4. Circular economy within the machinery revamping

5. Benefits of revamping for a greener economy

- Economic benefits
- Social benefits
- Environmental benefits

6. A more sustainable economic model - The Triple Bottom Line (TBL)

References

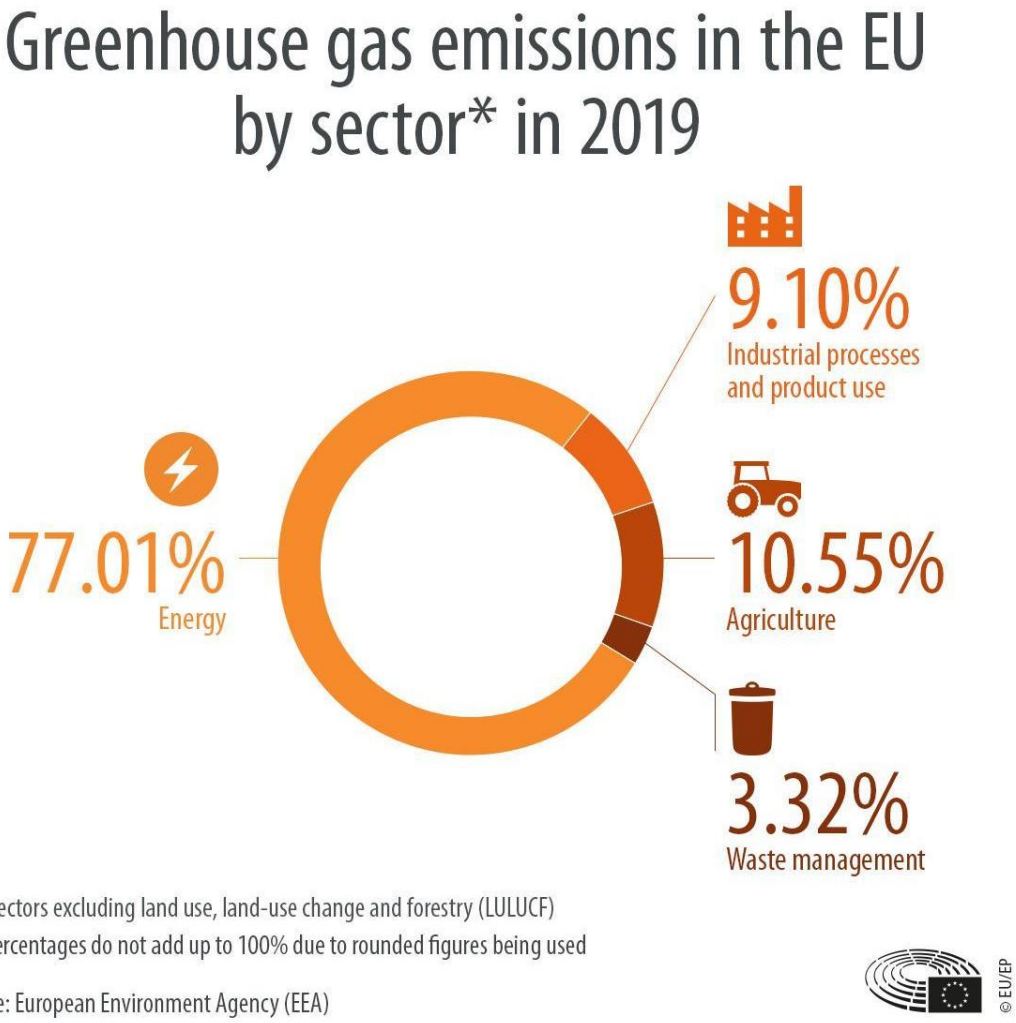
II. Interactive methods to develop Green Skills

Co-funded by the Erasmus+ Programme of the European Union

I. Theoretical part

1. Environmental sustainability and climate change

Climate change is already affecting Europe in various forms, depending on the region. It can lead to biodiversity loss, forest fires, decreasing crop yields and higher temperatures. It can also affect people's health.
In 2021, the EU made climate neutrality, the goal of zero net emissions by 2050, legally binding in the EU. It set an interim target of 55% emission reduction by 2030 with the ***European Green Deal***, this goal of zero net emissions is enshrined in the climate law. The European Green deal is the roadmap for the EU to become, climate-neutral by 2050.



The ***Climate Pact***, the movement of people united taking steps to build a more sustainable Europe, launched by the European Commission as part of the European Green Deal, will continue the EU's work in this area and actively support labor organizations, educational bodies, and public authorities to help those seeking employment in the green economy.

Co-funded by the Erasmus+ Programme of the European Union

In order to tackle pressing environmental challenges like climate change, pollution and plummeting biodiversity, nations and businesses need to transition towards greener, resilient and climate-neutral economies and societies.
For this reason, the International Labour Organization ILO created the ***Guidelines for a just transition toward environmentally sustainable economies and societies for all***.

A Just Transition means greening the economy in a way that is as fair and inclusive as possible to everyone concerned, creating decent work opportunities and leaving no one behind.
A Just Transition involves maximizing the social and economic opportunities of climate action while minimizing and carefully managing any challenges – including through effective social dialogue among all groups impacted, and respect for fundamental labor principles and rights.
Ensuring a just transition is important for all countries at all levels of development. It is also important for all economic sectors – by no means limited to energy supply – and in urban and rural areas alike.

[1] https://www.europarl.europa.eu/news/en/headlines/priorities/climate-change/20180703STO07129/eu-responses-to-climate-change

2. Towards a greener economy

A properly functioning circular economy does not only depend on the government and industry, consumers, too, have a role to play by choosing sustainable products, using them longer, repairing them or recycling them at the end of their life cycle.

The **circular economy** is an economy model that aims to minimize the withdrawal of resources, the biological ones, by reintegrating them into the biosphere and the technical ones by prolonging their use, promoting their reuse and putting them back into the cycle with recycling, thus minimizing production and waste disposal. The circular economy aims to replace the linear operating model of the traditional economy which is based on the massive withdrawal of natural resources, their transformation into products that are consumed, generating large quantities of waste that are disposed of.

The **green economy** aims to save and efficiently use resources and energy, on the development of renewable energy, recycling and renewal of materials in order to have better quality inclusive well-being, protecting natural capital and eco-systemic services.
The green economy is a vision of the economy in the era of the global climate crisis and environmental scarcity which therefore considers the ecological question a decisive driver for the possibilities of development, better well-being and social inclusion that takes into account not only a more equitable distribution of goods, but also of the damages caused to natural capital and ecosystem services.
The circular economy can be considered the pillar of a green economy.

The transition to a climate-neutral economy will trigger a fundamental transformation across a wide range of sectors. New jobs will be created, while some jobs will be replaced and others redefined.

It’s becoming necessary to:

Co-funded by the Erasmus+ Programme of the European Union

- promote and support green employment
- address the skilling and reskilling of workers
- anticipate changes in workplaces of the future

For this reason, **green skills** are now a requirement for accessing the most diverse professions, because of the great importance that environmental issues are acquiring even within the production sector.

[2] https://europa.eu/climate-pact/about/priority-topics/green-skills_en

3 https://www.ilo.org/global/topics/green-jobs/WCMS_824102/lang--en/index.htm

3. What are green skills

According to the definition of UNIDO - the United Nations organization for industrial development, **Green Skills** are the *knowledge, abilities, values and attitudes needed to develop and support a sustainable and resource-efficient society*.

Green Skills today means all those skills that allow us to respond to the need for sustainable reconversion of production in every type of reality, from public and private offices to shops, industries, and companies.

Green skills can be summarized in two main areas:
- predisposition to energy saving
- attitude to environmental sustainability

The demand for green skills is now transversal and concerns all professions. An increasing number of companies are looking for professional profiles able to work with tools and products related to eco-sustainability. There are entire production sectors such as sustainable tourism, sustainable construction and mechatronics where the foundations of the Green Economy have a great impact.

Companies evaluate, as green skills for work, all those factors such as the attitude to energy saving and environmental sustainability, so they turn to human resources that demonstrate ability, skills, and attention in making corporate activities more environmentally friendly. Among the new trends that change the labor market, there is not only the creation and/or activation of new green jobs, in fact, but the attitude to energy saving and environmental sustainability is also the first skill required by companies immediately after the so-called soft skills.
Green skills are presented both as a specific technical skill and as a personal orientation and a cultural propensity. They not only concern the ability to set up the technological renewal of production chains, or greater energy or water efficiency but are also assessed on the basis of the ability to change individual and organizational behaviors on the objectives of eco-sustainability within the production processes.

[4] https://www.fondazionesviluppososteni bile.org/circular-economy-pilastro-green-economy/

[5] https://www.unido.org/stories/what-are-green-skills

[6] https://jobspa.it/blog/competenze-green-piu-richieste-in-futuro

Why green jobs are essential for the future (video)

4. Circular economy within the machinery revamping

The "green professions" include both specific professions, which are required to meet the new needs of the Green Economy, and those that will have to face the challenge of reskilling skills in a green key.
The professions in which these skills are most required are: civil engineers, electronic and telecommunications engineers, construction site management technicians, occupational safety technicians, energy and mechanical engineers.

The hope of European companies is that positions related to the reduction of environmental impacts in the technological-digital field will develop.
As an example, in the most structured companies, it can be found the position of the mobility manager: a person in charge of corporate mobility, who is responsible for coordinating employee travel from home to work in a more sustainable way .

The fundamental principle on which the economic model of the circular economy is based, which aims to minimize the withdrawal of resources by promoting the reuse of biological resources by reintegrating them into the biosphere and of the technical ones by prolonging their use, is perfectly suited to the refitting sector and machinery revamping.
The concept of circular economy is the basis for explaining some of the benefits of refurbishing obsolete machinery. This covers the economic as well as the social and environmental aspects.

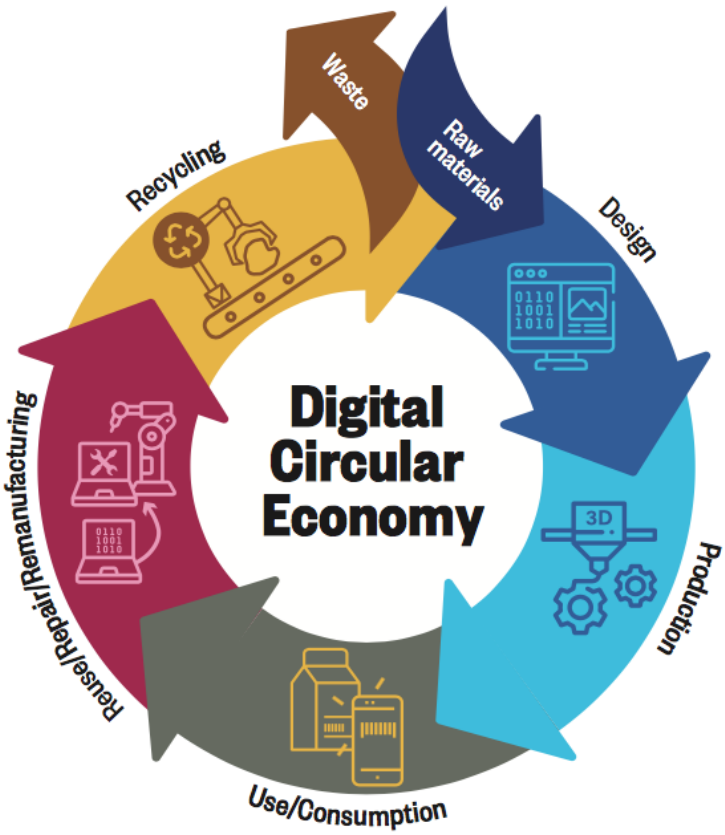
Co-funded by the Erasmus+ Programme of the European Union

The adoption of the circular systems in the industry consists of reuse, sharing, repair, revamp, refurbishment, remanufacturing and recycling to create a close-loop system, minimizing the use of resource inputs and the creation of waste, pollution and carbon emissions. Many of these aspects are included in the machinery revamping.

The recovery of obsolete machinery will have a strong influence on this whole model as it responds not only to environmental programs for waste management, energy efficiency and raw material reduction, but will also contribute decisively to the circular economy.

The machinery revamping brings positive financial and economic benefits. By raising the productivity of industry, modernized machines also raise the overall production of the economy; as a result, employment, national income, and the growth rate of the economy increase.

Revamping is generally cheaper than buying new products or equipment. For example, when revamping a vehicle, functional parts are reused (like the seats, chassis, gears, windows, etc.). This not only increases safety due to using more advanced tech solutions, but also reduces operating costs.
Also, revamping or retrofitting are common practices that embrace key principles of the “circular economy”, whereby machines are designed with ease of maintenance in mind and with much of the materials to be recycled at the end of their lifetime. Spare parts of old/obsolete machines are repaired or reproduced for more effective use, thus reducing the amount of waste and expanding the lifetime of the product while using fewer resources. These practices are key aspects in improving the environmental performance of machine tools as well, while not ensuring customer demands for productivity and reliability.

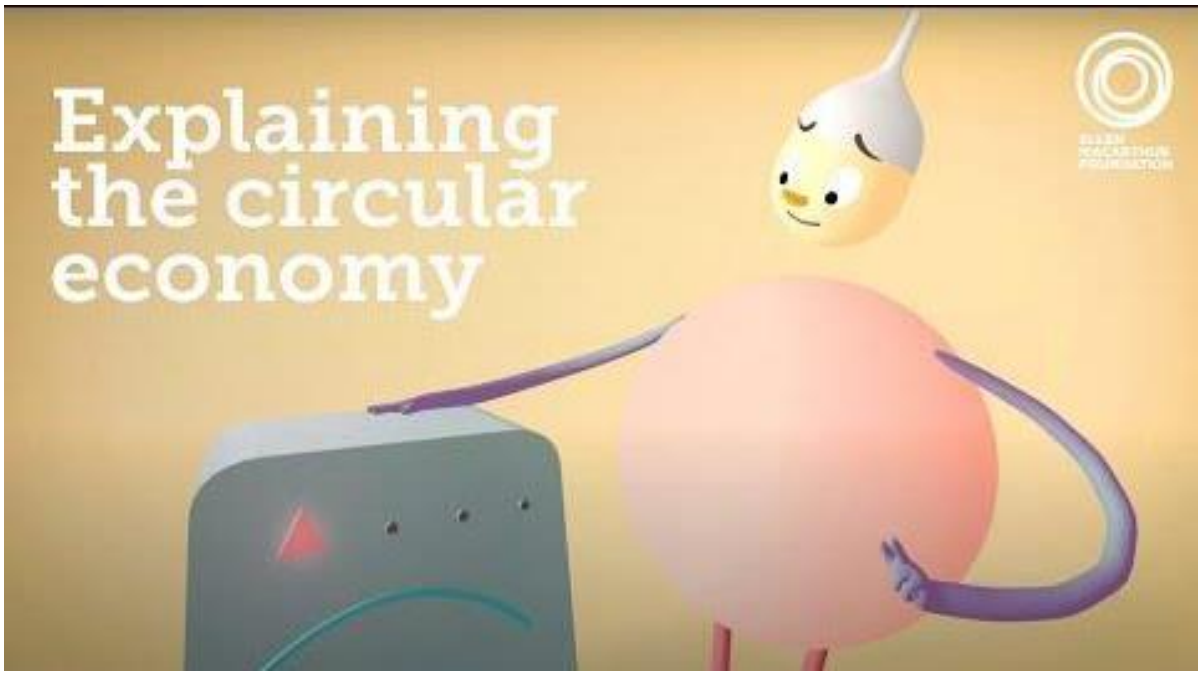


Source https://www.climate-kic.org/wp-content/uploads/2019/07/DRCE.pdf

Co-funded by the Erasmus+ Programme of the European Union

Machinery revamping is one of the key elements of circular economy. The companies that are using this system are able to retail and recycle the machines or their parts and modernize their equipment. In this way, companies will reduce the costs, update the machines according to their needs and follow the method of the digital circular manufacturing and modernizing of machines.

What If We Don't Buy Products and We Buy Service? Circular Economy Explained (Video)



[7] https://excelsior.unioncamere.net/index.php?option=com_content&view=article&id=349:le-competenze-green&

[8] https://laborability.com/approfondimenti/leuropa-alla-ricerca-di-competenze-green-e-digital

[9] https://blog.ener2crowd.com/circular-economy-e-green-economy/

5. Benefits of revamping for a greener economy

Economic benefits

- **Update of the software and/or hardware**
  The installation of a new software or hardware could be very expensive for the companies. Updating the current software/hardware installations costs are reduced without bearing the expenses for a new software version.

- **Increase in productivity**
  Replacing outdated machines can be quite expensive. For companies, revamping is the best solution to update their machines and systems according to their needs at the lowest cost.
  The use of machines after revamping helps producers to increase their income because a piece of updated machinery will operate faster. Users, operating a computer or one part of the new machine, will be able to complete the work faster than many workers engaged in doing the same work manually.

Co-funded by the Erasmus+ Programme of the European Union

- **Increased efficiency of the worker**
  By increasing the efficiency of workers, they can perform their duties in a better way than they would do manually. In this way, they will produce more accurately and faster qualitative products in larger quantities, moreover workers with more skills and competencies will be able to also increase their income.

- **Create employment opportunities**
  Machinery creates employment and increases productivity, reducing costs for the industry and making goods and products cheaper; this leads to demand increase. The industry needs more workers in order to face the demand.
  Some of the categories of increasing demand are Mechanical Engineers, Aerospace Engineering and Operations Technicians, Electro-mechanical Technicians, Sales Engineers, Computer and Information Research Scientists, Computer Programmers.

Social benefits

A need to update or "revamp" such machines, without wasting resources for buying new equipment, results in a variety of benefits for businesses and the society itself.
In fact, revamping or "retrofitting" has become a highly significant approach in achieving sustainability at all social, economic, and environmental levels, and improve people's standard of living.

- Reusing machinery components makes the process of modernization cheaper and more attractive, it also optimizes energy performance and help to prolong the life of machines

- Revamping pieces of machinery creates educational opportunities. It requires educated and skilled manpower for their operations, repairs, maintenance, and modernization. This leads to the demand of formal or non-formal technical education, which in turn creates a demand for relevant teaching staff. Educational opportunities regarding revamping extend to a large number of education fields, for example, engineering, machine learning, 3D design, software development, etc.

- Replacing or renewing outdated components of a machine, which results in better effectiveness of the entire machine, also results in a better workplace environment. Revamping ensures that a machine functions according to the latest technology demands and current standards of practices. In this way, the everyday work of employees dealing with machines and construction work is improved, including their working conditions and safety aspects.

- Revamping creates employment. Although a general notion regarding the machine industry tends to hold that machines replace the human workforce, a lot of human creativity is still needed when modernizing machines. when updating a machine, there are many levels that must be conserved: structure, dimensions, security systems etc. This turns revamping into an almost manual technique, unique to each brand and context, the need to Workers are always needed to manufacture new machines, or maintain, repair, and modernize older ones.

Co-funded by the Erasmus+ Programme of the European Union

Environmental benefits

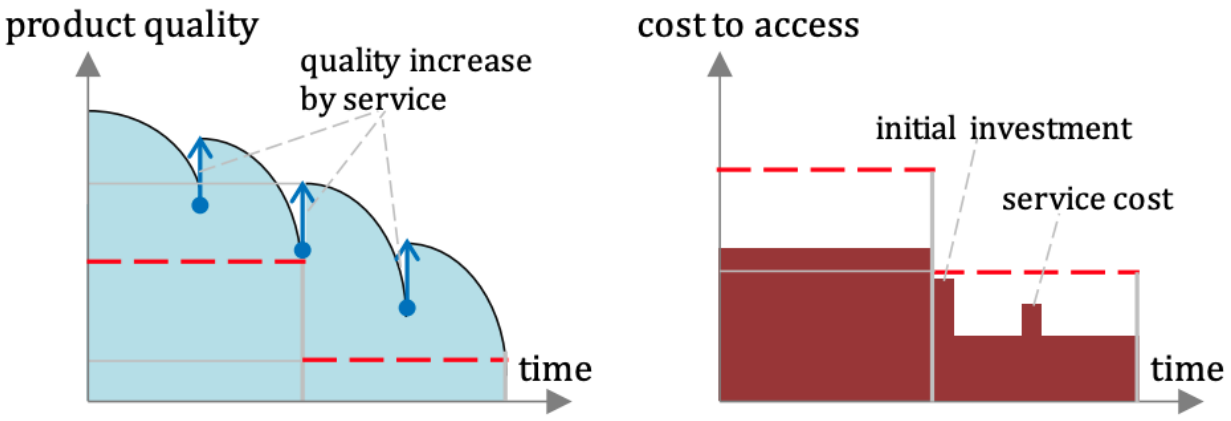
Today environmental sustainability is an important part of any economic activity and particularly in the field of industry.

The recovery of obsolete machinery will have a strong influence on this whole model as it responds not only to environmental programs for waste management, energy efficiency and raw material reduction, but will also contribute decisively to the circular economy.

One of the biggest challenges that industry is facing today is the need to further improve its **environmental performance** in order truly to become compatible with sustainable development. The industry must be an active actor in the process through responsible entrepreneurship and eco-efficiency. Increased environmental performance will mean reducing the negative environmental impacts that occur at each stage of the product life-cycle, from the extraction of raw materials through the production processes, transport and distribution of products to the use and disposal of products.
We must think about the need to promote a more sustainable model of economic activity that meets the double challenge of reducing its emissions and using its resources efficiently.
In this way, the model will move from being part of the problem to being part of the solution.

When we talk about sustainability, we do not refer exclusively to environmental issues, such as energy efficiency or climate change. The principle of sustainability is based on the connections between the environment, society and the economy.

Machinery revamping ensures that machines function smoothly according to modern technology demands. When modern equipment is up-to-date, through incorporating the latest technologies and features, they work more efficiently and are more probably to eliminate potential faults.
As a result, state-of-art components used for revamping results in the expand of the performance capacities of machines and leads to an overall more efficient workplace, that is safer and user-friendly for workers and employees, while being profitable for the employers.



Modernized machines help to increase the production and durability of goods and products. By incorporating state-of-art technologies to existing machinery, old machines become more productive and last longer, allowing mass and automate production of the products, and thus leading to large-scale production, reducing of costs, and raising of profits.

Co-funded by the
Erasmus+ Programme
of the European Union

6. A more sustainable economic model - The Triple Bottom Line (TBL)

In 1994, author and entrepreneur, John Elkington, built upon the concept of the **triple bottom line** (TBL) in hopes to transform the current financial accounting-focused business system to take on a more comprehensive approach in measuring impact and success. Historically, businesses operated in service solely to their financial bottom line. However, as a result of the triple bottom line theory and application, some businesses began to realize the connection among environmental health, social well-being and the organization's financial success and resilience.

Triple bottom line theory expands business success metrics to include contributions to environmental health, social well-being, and a just economy. These bottom-line categories are often referred to as the **three "P's"**: **people**, **planet**, and **prosperity**.

The rationale of this theory consisted in encouraging companies to operate in the reference economic context through strategies and decisions that were able to simultaneously enhance (i) the environment (planet), (ii) the social context (people) and (iii) the economic-financial aspect (profit). These were three elements which, if jointly considered by the company, would have allowed the creation of greater production value, while operating, at the same time, greater attractiveness for investors and consumers and favoring an environment more consciously oriented towards ecological and social sustainability between employees.

[10] https://sustain.wisconsin.edu/sustainability/triple-bottom-line/

Image Source https://www.researchgate.net/figure/The-interconnection-of-the-elements-of-the-Triple-Bottom-Line-concept_fig1_329185478

Co-funded by the Erasmus+ Programme of the European Union

- **People**

  This bottom line measures businesses' impact on human capital. A company using the triple bottom line has a responsibility to not only shareholders but also employees, vendors, customers, the community where it does business and anyone else impacted by the organization, whether directly or indirectly. It recognizes the interdependency of all the human relationships and interactions that enable the company to operate. This can translate into actions such as providing quality healthcare benefits and flexible work schedules to employees, offering opportunities for professional or educational advancement, creating a safe work environment, and engaging in fair labor practices.

- **Planet**

  Companies following the TBL model work to reduce their ecological footprint. They recognize that the smaller environmental impact a company has, the longer it can operate. At its most basic level, this involves not producing products that are unsafe or unhealthy for the planet and the people on it, but it also includes reducing consumption, waste and emissions. It involves specific actions, such as using renewable energy sources, reducing energy use, disposing of toxic materials safely and adopting a host of green corporate policies.

- **Profit**

  All companies are concerned about their financial standing, but businesses committed to the triple bottom line look at profits in terms of not just what they can do for shareholders, but also how they can help the broader community. In this model, a company helps stimulate economic growth and create wealth by compensating employees fairly, supporting local suppliers with its business, generating innovation, and paying its fair share of taxes. It also makes financially prudent but ethically driven decisions about how and where <to source materials, products or labor.

  In recent years, mainly due to the growing interest in environmental and social issues and in order to identify a criterion that was - even more than the Triple Bottom Line - capable of evaluating an investment as socially responsible, they have been elaborated by economic doctrines more avant-garde the so-called ESG (Environmental, Social, Governance) factors. Today, banking and financial institutions, and even more specialized organizations such as ESG rating agencies, are increasingly using the aforementioned paradigm as a yardstick to guide investment choices and the allocation of capital. From this it follows that a company, for example, to access certain forms of financing or public incentives, must necessarily make sustainable and responsible investments, respectful of environmental and social aspects, as well as aimed at generating profits.

  The environmental and social components play an increasingly predominant role also in the choices of consumers who prefer companies that carry out their business in a sustainable and responsible way. This creates a strong link between socially responsible investments and corporate reputation. ESG factors as a guiding criterion for identifying corporate strategies and policies contribute significantly to improving and increasing the company's reputation among investors and consumers.

  The growing protection of the environment and respect for human capital are two aspects that, today in particular, require specific attention in any area. Thanks to the Triple Bottom Line concept and ESG factors, social and environmental issues play a role of increasing importance in the economic sector, and their evaluation represents an essential step in the sustainable and responsible investment strategies of companies.

Co-funded by the Erasmus+ Programme of the European Union

[11] https://www.previti.it/dal-concetto-di-triple-bottom-line-ai-fattori-esg-le-nuove-politiche-globali
[12] www.business.com/articles/triple-bottom-line/
[13] https://www.previti.it/dal-concetto-di-triple-bottom-line-ai-fattori-esg-le-nuove-politiche-globali

References

- https://www.europarl.europa.eu/news/en/headlines/priorities/climate-change/20180703STO07129/eu-responses-to-climate-change
- https://europa.eu/climate-pact/about/priority-topics/green-skills_en
- https://www.ilo.org/global/topics/green-jobs/WCMS_824102/lang--en/index.htm
- https://www.fondazionesvilupposostenibile.org/circular-economy-pilastro-green-economy/
- https://www.unido.org/stories/what-are-green-skills
- https://jobspa.it/blog/competenze-green-piu-richieste-in-futuro
- https://excelsior.unioncamere.net/index.php?option=com_content&view=article&id=349:le-competenze-green&
- https://laborability.com/approfondimenti/leuropa-alla-ricerca-di-competenze-green-e-digital
- https://blog.ener2crowd.com/circular-economy-e-green-economy/
- https://sustain.wisconsin.edu/sustainability/triple-bottom-line/
- https://www.previti.it/dal-concetto-di-triple-bottom-line-ai-fattori-esg-le-nuove-politiche-globali
- www.business.com/articles/triple-bottom-line/
- https://www.previti.it/dal-concetto-di-triple-bottom-line-ai-fattori-esg-le-nuove-politiche-globali

Co-funded by the Erasmus+ Programme of the European Union

II. Interactive methods to develop Green Skills

1. Example of digital green skills practical application: Retrofitting of an old FM Radio (case study)

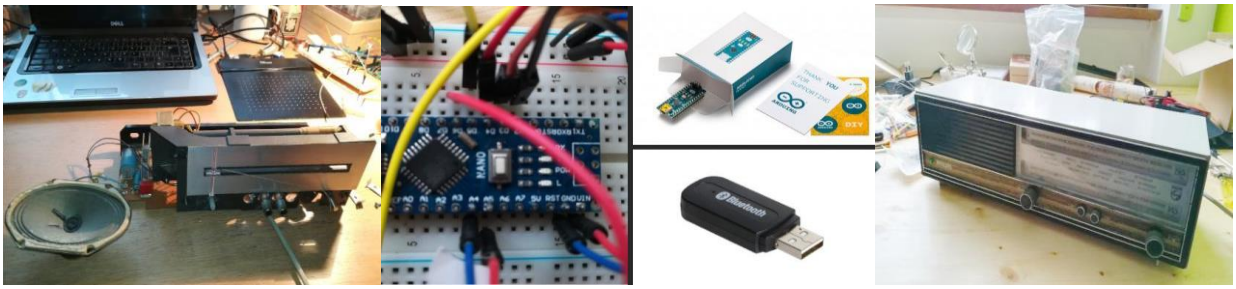
Green Skills today means all those skills that allow us to respond to the need for sustainable reconversion of production in every type of reality, from public and private offices to shops, industries, and companies.

Green skills can be summarized in two main areas:
- predisposition to energy saving
- attitude to environmental sustainability

Precisely in the context of environmental sustainability, lies the need to apply green skills to retrofit old machinery and objects with the intention of giving them new life and a new cycle of use, avoiding the purchase of a new product that would feed the intrinsic mechanism of the linear classic economic model.

A practical example for developing green skills in the digital field, can be seen in the case study about the Retrofitting of an old FM radio providing Bluetooth connectivity through an ARDUINO microcontroller.

The model used is the Arduino Nano which controls the turning on of the Bluetooth stick and the operation of the FM radio module. Through Arduino you can then select whether to operate the radio or Bluetooth through a button for the switch. In this way our radio can be used both to listen to the FM radio and to listen to music from our smartphone.



Author: Giulio Pons
Source: https://hackaday.io/project/162367-retrofitting-of-an-old-fm-philips-radio
https://www.youtube.com/watch?v=MuXsjAEGdNI

[1] https://jobspa.it/blog/competenze-green-piu-richieste-in-futuro

Co-funded by the Erasmus+ Programme of the European Union

2. Promote and develop Green Skills through Gaming: Penji protects the planet! (online game)

Penjii Protects the planet is a mobile game that aims to teach players how to save their planet! This game is developed by Caped Koala Studios a result of an Erasmus+ project "Promoting Green Skills Through Games" project with partners from Austria, Croatia, Ireland and Spain all working together to promote Green Skills.

This game is an endless runner-style game, where the penguin Penjii runs around the planet to combat the problems facing our planet.

Penjii visits 4 different parts of the world (the Antarctic, a Beautiful Beach, a Chaotic City, Your Home) and each level aims to teach players about a specific problem facing our environment and planet.

https://capedkoala.com/penjii-protects-the-planet/

3. Develop Green Skills through learning by doing: The renewable power of green skills for women in Zambia (video)

The women in the Kalulushi compound in the Copperbelt Province, built their own houses with green technologies. With the help of the ILO through the Zambia Green Jobs Program and the Zambia Homeless and Poor people Federation, they got a loan to buy a small plot of land and they were trained in green technologies skills learning by doing to build houses using bricks made by sustainable materials.

https://www.youtube.com/watch?v=4K3AiZaiMoc

Co-funded by the
Erasmus+ Programme
of the European Union